Patent Assignment Abstract of Title

Total Assignments: 2

Application #: 09723855 **Filing Dt:** 11/27/2000 **Patent #:** NONE **Issue Dt:**

PCT #: NONE **Publication #:** NONE **Pub Dt:**

Inventors: Jonathan James Oliver, Wray Lindsay Buntine, George Roumeliotis

Title: System and method for adaptive text recommendation

Assignment: 1

Reel/Frame: 011294/0912 **Received:** 12/06/2000 **Recorded:** 11/28/2000 **Mailed:** 02/07/2001 **Pages:** 2

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors: OLIVER, JOHANTHAN J. **Exec Dt:** 11/22/2000

BUNTINE, WRAY LINDSAY **Exec Dt:** 11/22/2000

ROUMELIOTIS, GEORGE **Exec Dt:** 11/22/2000

Assignee: DYNAPTICS CORPORATION
SUITE 400
TWO NORTH SECOND STREET
SAN JOSE, CALIFORNIA 95113

Correspondent: FERNANDEZ & ASSOCIATES
DENNIS FERNANDEZ
P.O. BOX D
MENLO PARK, CA 94026-6204

Assignment: 2

Reel/Frame: 011688/0993 **Received:** 04/20/2001 **Recorded:** 04/09/2001 **Mailed:** 06/26/2001 **Pages:** 3

Conveyance: RE-RECORD TO CORRECT THE NAME OF THE CONVEYING PARTY, PREVIOUSLY RECORDED ON REEL 011294 FRAME 0912, ASSIGNOR CONFIRMS THE ASSIGNMENT OF THE ENTIRE INTEREST.

Assignors: OLIVER, JONATHAN J. **Exec Dt:** 11/22/2000

BUNTINE, WRAY LINDSAY **Exec Dt:** 11/22/2000

ROUMELIOTIS, GEORGE **Exec Dt:** 11/22/2000

Assignee: DYNAPTICS CORPORATION
TWO NORTH SECOND STREET, SUITE 400
SAN JOSE, CALIFORNIA 95113

Correspondent: FERNANDEZ & ASSOCIATES, LLP
DENNIS FERNANDEZ, ESQ.
PATENT ATTORNEYS
PO BOX D
MENLO PARK, CALIFORNIA 94026-6204

Search Results as of: 5/6/2003 2:23:44 P.M.

	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L5	4134	text$1 adj document$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 15:48
2	BRS	L6	138	5 and 707/10.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 14:52
3	BRS	L7	42	6 and (receiv$4 same quer$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 14:56
4	BRS	L8	4	7 and (extract$4 same (keywords$1 or (key adj word$1)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 14:57
5	BRS	L9	0	8 and adapt$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 14:55
6	BRS	L10	527	5 and (receiv$4 same quer$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 15:48
7	BRS	L11	52	10 and (extract$4 same (keywords$1 or (key adj word$1)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 14:57
8	BRS	L12	11	11 and filter$4 and adapt$5 and cluster$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 15:06
9	BRS	L13	32	11 and (filter$4 or hierarch$5 or index$4) and adapt$5 and (cluster$4 or group$4 or class$6)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 15:26

	Type	L #	Hits	Search Text	DBs	Time Stamp
10	BRS	L14	2	60/237795	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 15:27
11	BRS	L15	22013	text$1 same document$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 15:48
12	BRS	L16	755	15 and (receiv$4 same quer$3) and (filter$4 or hierarch$5 or index$4) and adapt$5 and (cluster$4 or group$4 or class$6)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 16:05
13	BRS	L17	76	16 and 707/10.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 16:12
14	BRS	L18	8	17 and theme$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 16:16
15	BRS	L19	2	5794233.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/05/06 16:17

	Document ID	Issue Date	Title	Current OR	Current XRef	Retrieval Classif	Inventor
1	US 20020188606 A1	20021212	Organizing and accessing electronic business cards by virtual subdomain	707/10		707/10	Sun, Chen et al.
2	US 20020095454 A1	20020718	Communications system	709/201	707/10; 707/203; 707/204; 709/212; 709/228; 709/242; 709/244	707/10	Reed, Drummond Shattuck et al.
3	US 20020083054 A1	20020627	Scoping queries in a search engine	707/5	707/10	707/10	Peltonen, Kyle et al.
4	US 20020062310 A1	20020523	Peer-to-peer commerce system	707/3	707/10	707/10	Marmor, Michael et al.
5	US 20010049681 A1	20011206	Integrated multidimensional database	707/10	707/1; 707/104.1	707/10	Bova, G. Steven
6	US 20010044758 A1	20011122	Methods and systems for enabling efficient search and retrieval of products from an electronic product catalog	705/27	705/26; 707/10; 707/104.1; 707/200	707/10	Talib, Iqbal et al.
7	US 20010021935 A1	20010913	Network based classified information systems	715/513	707/10; 707/102	707/10	Mills, Dudley John
8	US 20010007086 A1	20010705	SYSTEM AND METHOD FOR DISTRIBUTED COMPUTER AUTOMOTIVE SERVICE EQUIPMENT	701/33	707/10	707/10	ROGERS, STEVEN W. et al.
9	US 6560596 B1	20030506	Multiscript database system and method	707/4	707/10; 707/100; 709/223	707/10	Margulies, Benson I. et al.
10	US 6553317 B1	20030422	Relational database and system for storing information relating to biomolecular sequences and reagents	702/20	435/6; 702/19; 707/10	707/10	Lincoln, Stephen E. et al.
11	US 6535881 B1	20030318	Distributed computer database system and method employing intelligent agents	707/10	707/104.1; 709/217	707/10	Baclawski, Kenneth P.
12	US 6529909 B1	20030304	Method for translating an object attribute converter in an information services patterns environment	707/10	707/1	707/10	Bowman-Amuah, Michel K.
13	US 6523021 B1	20030218	Business directory search engine	707/2	707/10; 707/104.1	707/10	Monberg, James C. et al.
14	US 6516312 B1	20030204	System and method for dynamically associating keywords with domain-specific search engine queries	707/3	707/1; 707/10; 707/2; 707/4; 707/5; 707/6	707/10	Kraft, Reiner et al.
15	US 6484156 B1	20021119	Accessing annotations across multiple target media streams	707/1	707/10	707/10	Gupta, Anoop et al.
16	US 6480843 B2	20021112	Supporting web-query expansion efficiently using multi-granularity indexing and query processing	707/5	704/7; 707/10	707/10	Li, Wen-Syan
17	US 6466933 B1	20021015	Delayed delivery of query results or other data from a federated server to a federated client until such information is needed	707/3	707/1; 707/10; 707/100; 707/4; 707/5	707/10	Huang, Mei-Ing W. et al.
18	US 6463433 B1	20021008	Distributed computer database system and method for performing object search	707/5	707/10; 707/104.1	707/10	Baclawski, Kenneth P.
19	US 6460038 B1	20021001	System, method, and article of manufacture for delivering information to a user through programmable network bookmarks	707/10	709/229; 709/245	707/10	Khan, Umair et al.
20	US 6457002 B1	20020924	System and method for maintaining a knowledge base and evidence set	707/3	707/10; 707/5; 707/6	707/10	Beattie, Thomas W. et al.

	Document ID	Issue Date	Title	Current OR	Current XRef	Retrieval Classif	Inventor
21	US 6442549 B1	20020827	Method, product, and apparatus for processing reusable information	707/10	705/21; 707/100; 709/201; 709/203; 709/217; 709/218	707/10	Schneider, Eric
22	US 6438539 B1	20020820	Method for retrieving data from an information network through linking search criteria to search strategy	707/3	707/10	707/10	Korolev, Anatoly Y. et al.
23	US 6434568 B1	20020813	Information services patterns in a netcentric environment	707/103R	707/10; 709/203	707/10	Bowman-Amuah, Michel K.
24	US 6430558 B1	20020806	Apparatus and methods for collaboratively searching knowledge databases	707/5	707/10	707/10	Delano, Paul A.
25	US 6405203 B1	20020611	Method and program product for preventing unauthorized users from using the content of an electronic storage medium	707/10	382/205; 705/26; 707/104.1; 709/203; 709/219	707/10	Collart, Todd R.
26	US 6401085 B1	20020604	Mobile communication and computing system and method	707/4	705/2; 707/10; 707/3; 709/223; 709/226	707/10	Gershman, Anatole Vitaly et al.
27	US 6397219 B2	20020528	Network based classified information systems	707/10	707/102	707/10	Mills, Dudley John
28	US 6385600 B1	20020507	System and method for searching on a computer using an evidence set	707/3	706/60; 707/10; 707/4; 709/218	707/10	McGuinness, Deborah L. et al.
29	US 6381602 B1	20020430	Enforcing access control on resources at a location other than the source location	707/9	707/1; 707/10; 709/217	707/10	Shoroff, Srikanth et al.
30	US 6374260 B1	20020416	Method and apparatus for uploading, indexing, analyzing, and searching media content	707/104.1	345/716; 707/10; 707/101; 707/3	707/10	Hoffert, Eric M. et al.
31	US 6370543 B2	20020409	Display of media previews	707/104.1	707/10; 725/113	707/10	Hoffert, Eric M. et al.
32	US 6366916 B1	20020402	Configurable and extensible system for deploying asset management functions to client applications	707/10	707/1	707/10	Baer, William J. et al.
33	US 6356905 B1	20020312	System, method and article of manufacture for mobile communication utilizing an interface support framework	707/10	705/26; 705/35; 707/102; 707/3; 707/5; 709/203; 709/219	707/10	Gershman, Anatole Vitaly et al.
34	US 6345288 B1	20020205	Computer-based communication system and method using metadata defining a control-structure	709/201	707/1; 707/10; 707/102; 707/104.1; 709/200; 709/203; 709/212; 709/216; 709/227; 709/229	707/10	Reed, Drummond Shattuck et al.
35	US 6338059 B1	20020108	Hyperlinked search interface for distributed database	707/4	707/10; 707/104.1; 715/500.1	707/10	Fields, Duane Kimbell et al.
36	US 6324566 B1	20011127	Internet advertising via bookmark set based on client specific information	709/203	345/854; 707/10; 707/104.1; 707/203; 707/204; 709/245; 709/247; 715/501.1	707/10	Himmel, Maria Azua et al.

	Document ID	Issue Date	Title	Current OR	Current XRef	Retrieval Classif	Inventor
37	US 6324538 B1	20011127	Automated on-line information service and directory, particularly for the world wide web	707/10	709/217; 709/218; 713/202	707/10	Wesinger, Jr., Ralph E. et al.
38	US 6314423 B1	20011106	Searching and serving bookmark sets based on client specific information	707/10	707/104.1; 709/218; 709/245; 715/513	707/10	Himmel, Maria Azua et al.
39	US 6314420 B1	20011106	Collaborative/adaptive search engine	707/3	707/10; 707/2; 707/5	707/10	Lang, Andrew K. et al.
40	US 6308175 B1	20011023	Integrated collaborative/content-based filter structure employing selectively shared, content-based profile data to evaluate information entities in a massive information network	707/10	707/1; 707/102; 707/2; 707/3; 707/5	707/10	Lang, Andrew K. et al.
41	US 6282549 B1	20010828	Indexing of media content on a network	707/104.1	707/1; 707/10; 707/101; 707/102; 707/103R; 707/2; 707/3; 715/500; 715/513	707/10	Hoffert, Eric M. et al.
42	US 6266668 B1	20010724	System and method for dynamic data-mining and on-line communication of customized information	707/10	706/15; 707/100	707/10	Vanderveldt, Ingrid V. et al.
43	US 6240412 B1	20010529	Integration of link generation, cross-author user navigation, and reuse identification in authoring process	707/5	345/854; 707/10; 707/100; 707/3; 707/4; 707/9; 709/203; 715/501.1; 715/511	707/10	Dyko, Denise Y. et al.
44	US 6223178 B1	20010424	Subscription and internet advertising via searched and updated bookmark sets	707/10	707/203; 709/218; 715/513	707/10	Himmel, Maria Azua et al.
45	US 6212522 B1	20010403	Searching and conditionally serving bookmark sets based on keywords	707/10		707/10	Himmel, Maria Azua et al.
46	US 6212516 B1	20010403	Parallel database management method and parallel database management system	707/3	345/866; 707/10; 707/8	707/10	Kobayashi, Susumu et al.
47	US 6192364 B1	20010220	Distributed computer database system and method employing intelligent agents	707/10	709/217; 709/218	707/10	Baclawski, Kenneth P.
48	US 6185608 B1	20010206	Caching dynamic web pages	709/216	707/10	707/10	Hon, Lenny K. et al.
49	US 6144964 A	20001107	Methods and apparatus for tuning a match between entities having attributes	707/10	707/6	707/10	Breese, John S. et al.
50	US 6134548 A	20001017	System, method and article of manufacture for advanced mobile bargain shopping	707/5	705/26; 707/10; 707/3; 709/217; 709/249	707/10	Gottsman, Edward et al.
51	US 6088717 A	20000711	Computer-based communication system and method using metadata defining a control-structure	709/201	707/10; 707/104.1; 707/203; 707/204; 709/212; 709/227; 709/229; 709/242; 709/244	707/10	Reed, Drummond Shattuck et al.

	Document ID	Issue Date	Title	Current OR	Current XRef	Retrieval Classif	Inventor
52	US 6044205 A	20000328	Communications system for transferring information between memories according to processes transferred with the information	709/201	707/1; 707/10; 707/102; 707/104.1; 709/200; 709/203; 709/212; 709/216; 709/229	707/10	Reed, Drummond Shattuck et al.
53	US 6038561 A	20000314	Management and analysis of document information text	707/6	707/10; 707/2; 707/3; 715/522	707/10	Snyder, David L. et al.
54	US 6029175 A	20000222	Automatic retrieval of changed files by a network software agent	707/104.1	707/10; 707/200; 707/201; 707/203; 709/202; 709/203	707/10	Chow, Yen-whei et al.
55	US 6029161 A	20000222	Multi-level mindpool system especially adapted to provide collaborative filter data for a large scale information filtering system	707/1	707/10; 707/102	707/10	Lang, Andrew K. et al.
56	US 6021409 A	20000201	Method for parsing, indexing and searching world-wide-web pages	707/102	707/10; 707/2	707/10	Burrows, Michael
57	US 5987464 A	19991116	Method and system for periodically updating data records having an expiry time	707/10	707/104.1	707/10	Schneider, Eric
58	US 5987454 A	19991116	Method and apparatus for selectively augmenting retrieved text, numbers, maps, charts, still pictures and/or graphics, moving pictures and/or graphics and audio information from a network resource	707/4	707/10; 707/101; 707/201; 715/501.1	707/10	Hobbs, Allen
59	US 5983228 A	19991109	Parallel database management method and parallel database management system	707/10	707/2	707/10	Kobayashi, Susumu et al.
60	US 5983214 A	19991109	System and method employing individual user content-based data and user collaborative feedback data to evaluate the content of an information entity in a large information communication network	707/1	707/10; 725/116	707/10	Lang, Andrew K. et al.
61	US 5974416 A	19991026	Method of creating a tabular data stream for sending rows of data between client and server	707/10	707/100; 707/3; 707/4	707/10	Anand, Thulusalamatom Krishnamurthi et al.
62	US 5974412 A	19991026	Intelligent query system for automatically indexing information in a database and automatically categorizing users	707/3	707/10; 707/102	707/10	Hazlehurst, Brian L. et al.
63	US 5915249 A	19990622	System and method for accelerated query evaluation of very large full-text databases	707/5	707/10; 707/9	707/10	Spencer, Graham
64	US 5913215 A	19990615	Browse by prompted keyword phrases with an improved method for obtaining an initial document set	707/10	707/3; 715/513	707/10	Rubinstein, Seymour I. et al.
65	US 5903892 A	19990511	Indexing of media content on a network	707/10	345/716; 707/104.1; 707/2	707/10	Hoffert, Eric M. et al.
66	US 5893091 A	19990406	Multicasting with key words	707/3	707/10; 707/104.1; 707/4; 709/206; 709/218	707/10	Hunt, Douglas et al.

	Document ID	Issue Date	Title	Current OR	Current XRef	Retrieval Classif	Inventor
67	US 5873076 A	19990216	Architecture for processing search queries, retrieving documents identified thereby, and method for using same	707/3	704/270.1; 704/9; 705/27; 705/30; 707/10	707/10	Barr, Thomas et al.
68	US 5867799 A	19990202	Information system and method for filtering a massive flow of information entities to meet user information classification needs	707/1	707/10	707/10	Lang, Andrew K. et al.
69	US 5864863 A	19990126	Method for parsing, indexing and searching world-wide-web pages	707/103R	707/10; 707/104.1; 707/3	707/10	Burrows, Michael
70	US 5862325 A	19990119	Computer-based communication system and method using metadata defining a control structure	709/201	704/270.1; 707/10; 707/203; 707/204; 709/212; 709/228; 709/242; 709/244	707/10	Reed, Drummond Shattuck et al.
71	US 5859972 A	19990112	Multiple server repository and multiple server remote application virtual client computer	709/203	707/10; 707/3; 707/4; 709/223	707/10	Subramaniam, Shankar et al.
72	US 5826261 A	19981020	System and method for querying multiple, distributed databases by selective sharing of local relative significance information for terms related to the query	707/5	707/1; 707/10; 707/2; 707/3	707/10	Spencer, Graham
73	US 5765028 A	19980609	Method and apparatus for providing neural intelligence to a mail query agent in an online analytical processing system	706/25	706/14; 706/50; 706/59; 707/10; 707/2; 707/3	707/10	Gladden, Paul Edward
74	US 5745899 A	19980428	Method for indexing information of a database	707/102	707/10; 707/104.1; 707/2; 707/3	707/10	Burrows, Michael
75	US 5706507 A	19980106	System and method for controlling access to data located on a content server	707/104.1	345/700; 345/716; 345/866; 707/10; 707/4; 707/9	707/10	Schloss, Robert Jeffrey
76	US 5694594 A	19971202	System for linking hypermedia data objects in accordance with associations of source and destination data objects and similarity threshold without using keywords or link-difining terms	707/6	707/10; 707/2; 707/3; 709/218; 715/501.1; 715/513	707/10	Chang, Daniel

US006314420B1

(12) **United States Patent**
Lang et al.

(10) **Patent No.: US 6,314,420 B1**
(45) **Date of Patent: *Nov. 6, 2001**

(54) **COLLABORATIVE/ADAPTIVE SEARCH ENGINE**

(75) Inventors: **Andrew K. Lang; Donald M. Kosak,** both of Pittsburgh, PA (US)

(73) Assignee: **Lycos, Inc.,** Waltham, MA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(h) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/204,149**

(22) Filed: **Dec. 3, 1998**

Related U.S. Application Data

(63) Continuation-in-part of application No. 08/627,436, filed on Apr. 4, 1996, now Pat. No. 5,867,799.

(51) **Int. Cl.**[7] **G06F 17/30**

(52) **U.S. Cl.** **707/3**; 707/10; 707/2; 707/5

(58) **Field of Search** 707/1, 10, 102, 707/3, 2, 5

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,019,961	*	5/1991	Addesso et al.	364/192
5,117,349	*	5/1992	Tirfing et al.	707/5
5,249,262	*	9/1993	Baule	395/66
5,471,610	*	11/1995	Kawaguchi et al.	707/4
5,537,586	*	7/1996	Amram et al.	707/3
5,544,049	*	8/1996	Henderson et al.	364/419.19
5,563,998	*	10/1996	Yaksich et al.	395/149
5,563,999	*	10/1996	Yaksich et al.	395/149
5,608,447	*	3/1997	Farry et al.	348/7
5,649,186	*	7/1997	Ferguson	707/10
5,842,199	*	11/1998	Miller et al.	707/2
5,867,799	*	2/1999	Lang et al.	707/1
5,983,214	*	11/1999	Lang et al.	707/1
6,006,222		12/1999	Culliss	707/5
6,014,665		1/2000	Culliss	707/5
6,029,161	*	2/2000	Lang et al.	707/1
6,078,916		6/2000	Culliss	707/5
6,182,068		1/2001	Culliss	707/5

OTHER PUBLICATIONS

Michael Persin, Document Filtering for Fast Ranking, Proceeding of the seventeenth annual international ACM–SIGIR conference on research and development in information retrieval, Jul. 6, 1994, pp. 339–348.*

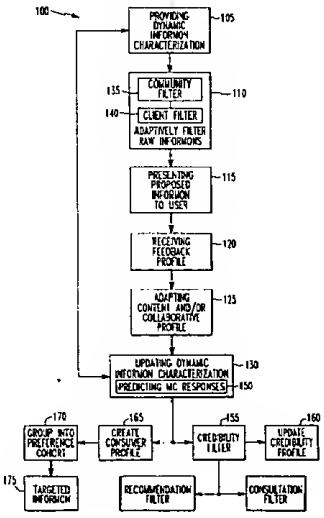
* cited by examiner

Primary Examiner—Thomas Black
Assistant Examiner—Frantz Coby
(74) *Attorney, Agent, or Firm*—Testa, Hurwitz & Thibeault, LLP

(57) **ABSTRACT**

A search engine system is provided for a portal site on the internet. The search engine system employs a regular search engine to make one-shot or demand searches for information entities which provide at least threshold matches to user queries. The search engine system also employs a collaborative/content-based filter to make continuing searches for information entities which match existing wire queries and are ranked and stored over time in user-accessible, system wires corresponding to the respective queries. A user feedback system provides collaborative feedback data for integration with content profile data in the operation of the collaborative/content-based filter. A query processor determines whether a demand search or a wire search is made for an input query.

36 Claims, 10 Drawing Sheets



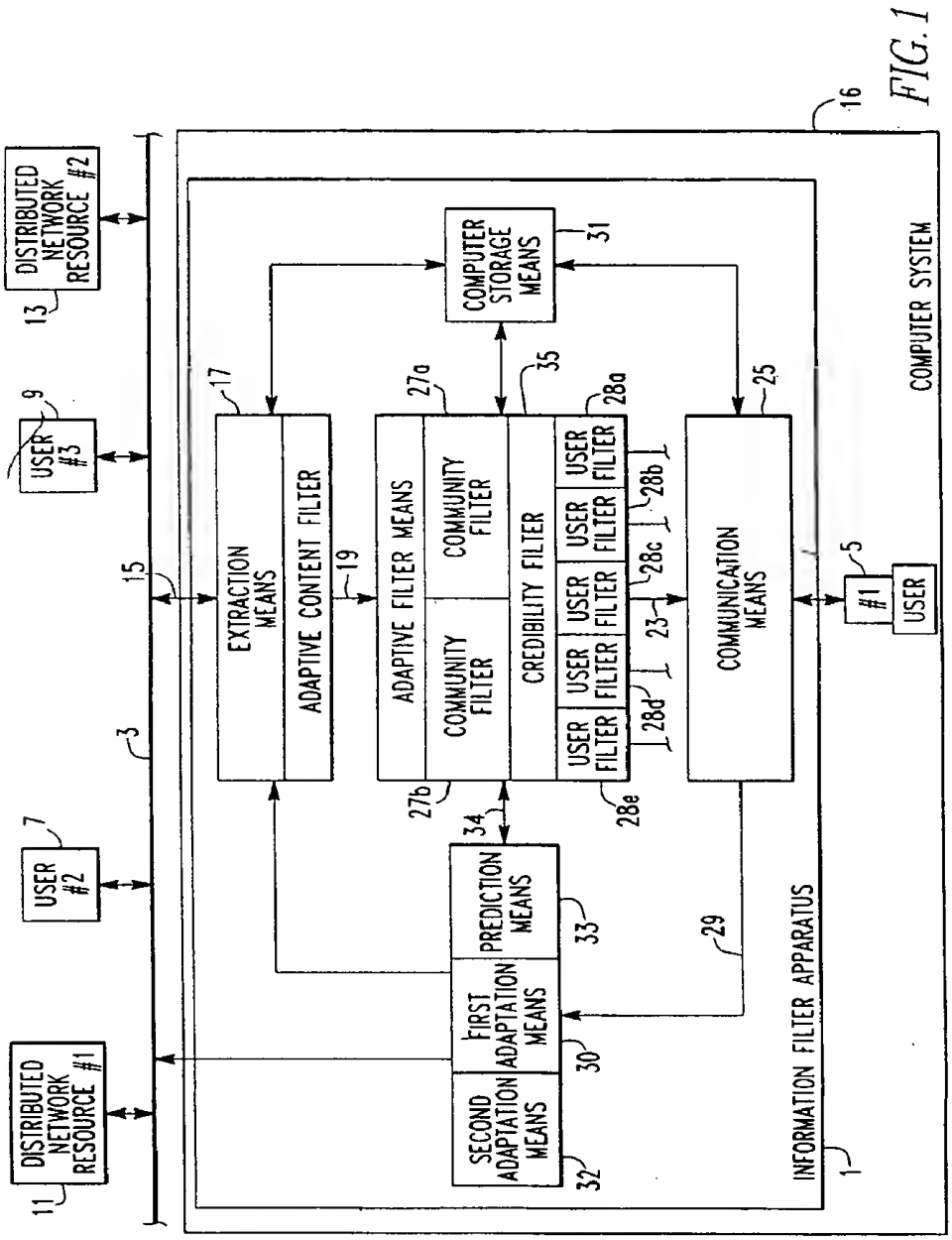

DISTRIBUTED NETWORK RESOURCE #2
13
USER #3
9
USER #2
7
DISTRIBUTED NETWORK RESOURCE #1
11
15
3
16
COMPUTER SYSTEM
INFORMATION FILTER APPARATUS
1
COMPUTER STORAGE MEANS
31
17
EXTRACTION MEANS
ADAPTIVE CONTENT FILTER
19
27a
ADAPTIVE FILTER MEANS
COMMUNITY FILTER
27b
COMMUNITY FILTER
35
CREDIBILITY FILTER
28a
USER FILTER
28b
28c
28d
28e
25
COMMUNICATION MEANS
23
5
#1 USER
34
PREDICTION MEANS
33
FIRST ADAPTATION MEANS
30
SECOND ADAPTATION MEANS
32
29

FIG. 1

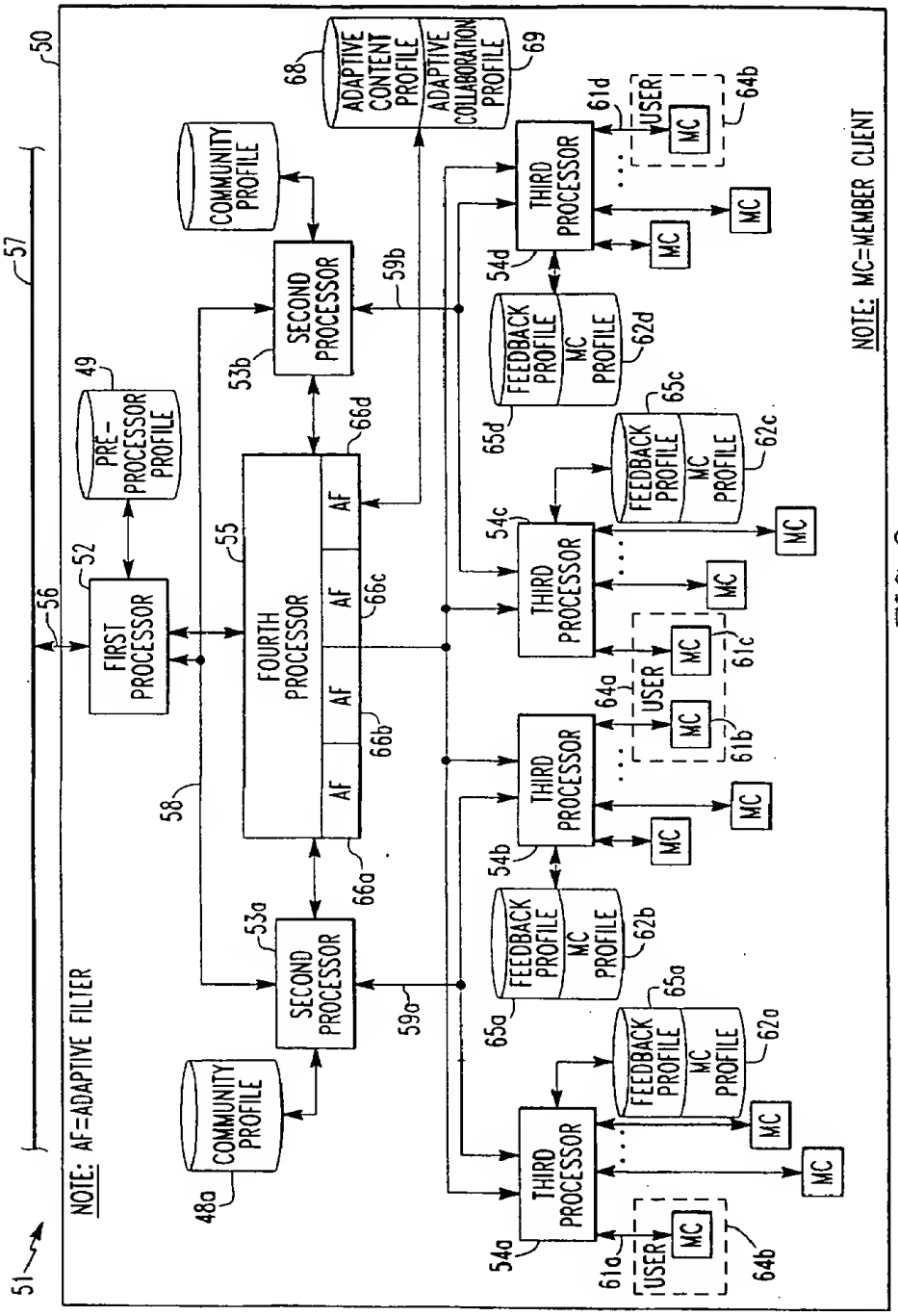

NOTE: AF=ADAPTIVE FILTER
NOTE: MC=MEMBER CLIENT
FIRST PROCESSOR
SECOND PROCESSOR
THIRD PROCESSOR
FOURTH PROCESSOR
AF
PRE-PROCESSOR PROFILE
COMMUNITY PROFILE
ADAPTIVE CONTENT PROFILE
ADAPTIVE COLLABORATION PROFILE
FEEDBACK PROFILE
MC PROFILE
USER
MC

FIG. 2

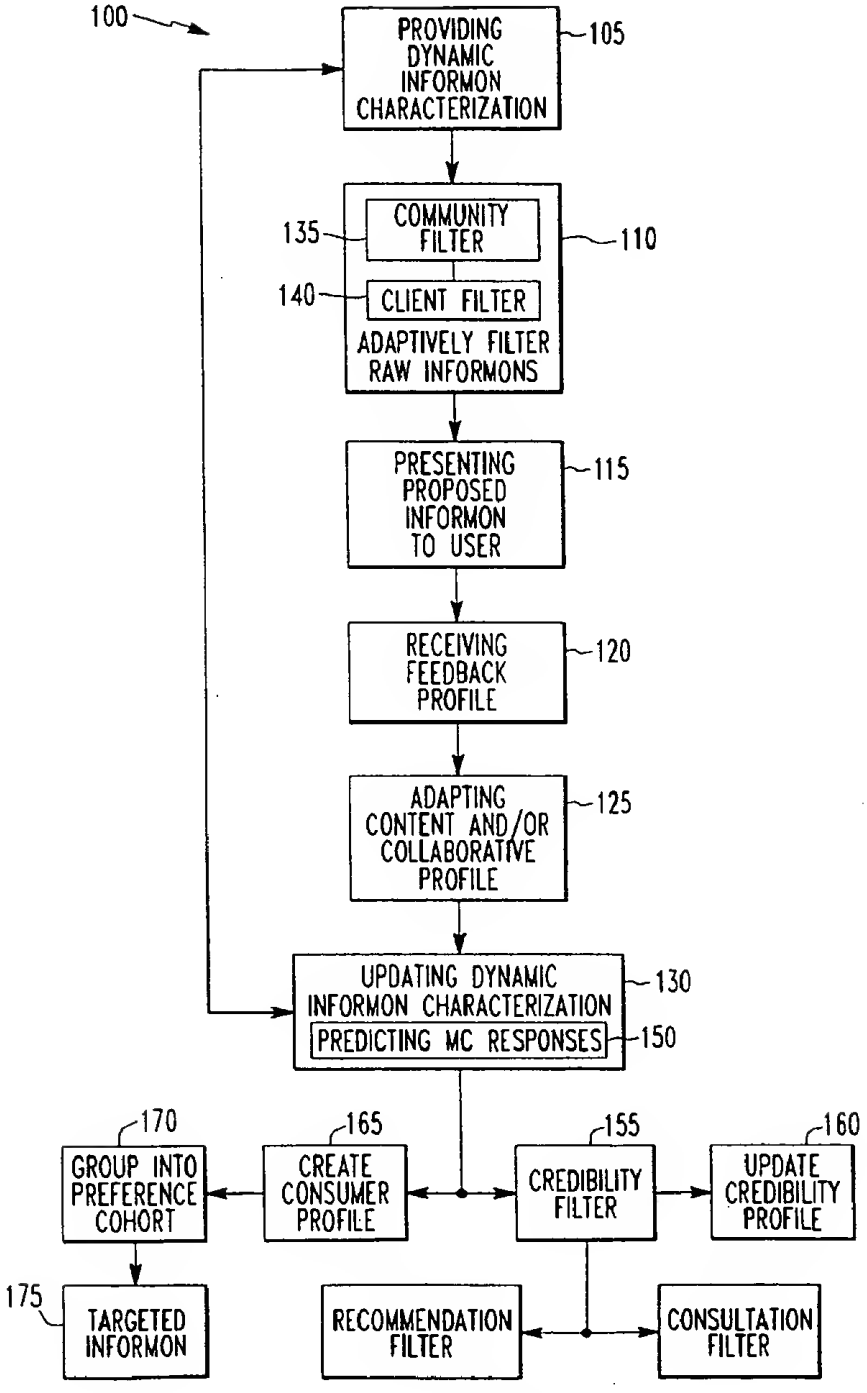

100
PROVIDING DYNAMIC INFORMON CHARACTERIZATION
105
COMMUNITY FILTER
135
CLIENT FILTER
140
110
ADAPTIVELY FILTER RAW INFORMONS
PRESENTING PROPOSED INFORMON TO USER
115
RECEIVING FEEDBACK PROFILE
120
ADAPTING CONTENT AND/OR COLLABORATIVE PROFILE
125
UPDATING DYNAMIC INFORMON CHARACTERIZATION
130
PREDICTING MC RESPONSES
150
170
GROUP INTO PREFERENCE COHORT
165
CREATE CONSUMER PROFILE
155
CREDIBILITY FILTER
160
UPDATE CREDIBILITY PROFILE
175
TARGETED INFORMON
RECOMMENDATION FILTER
CONSULTATION FILTER

FIG. 3

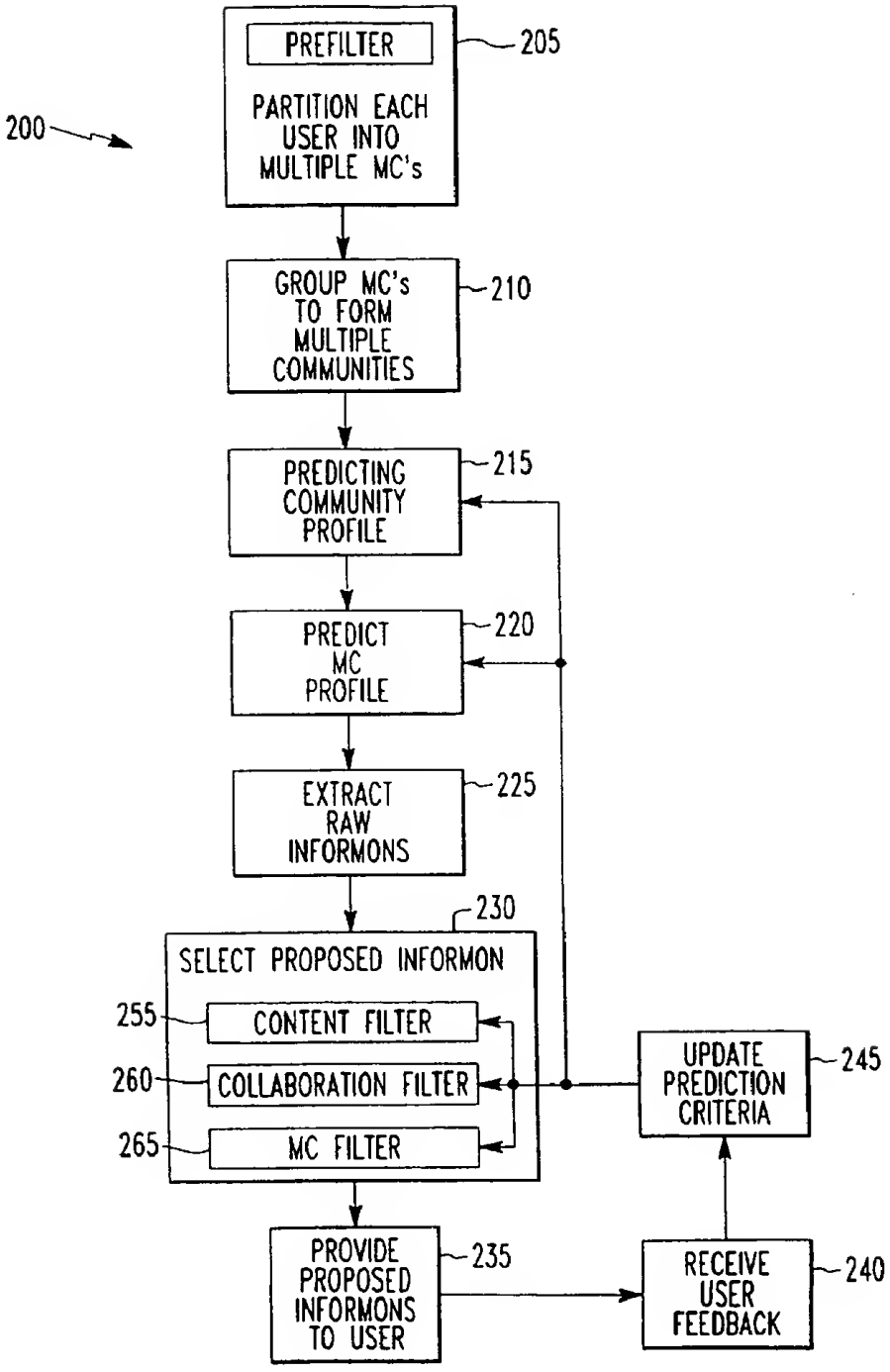

200
PREFILTER
PARTITION EACH USER INTO MULTIPLE MC's
205
GROUP MC's TO FORM MULTIPLE COMMUNITIES
210
PREDICTING COMMUNITY PROFILE
215
PREDICT MC PROFILE
220
EXTRACT RAW INFORMONS
225
230
SELECT PROPOSED INFORMON
255
CONTENT FILTER
260
COLLABORATION FILTER
265
MC FILTER
UPDATE PREDICTION CRITERIA
245
PROVIDE PROPOSED INFORMONS TO USER
235
RECEIVE USER FEEDBACK
240

FIG. 4

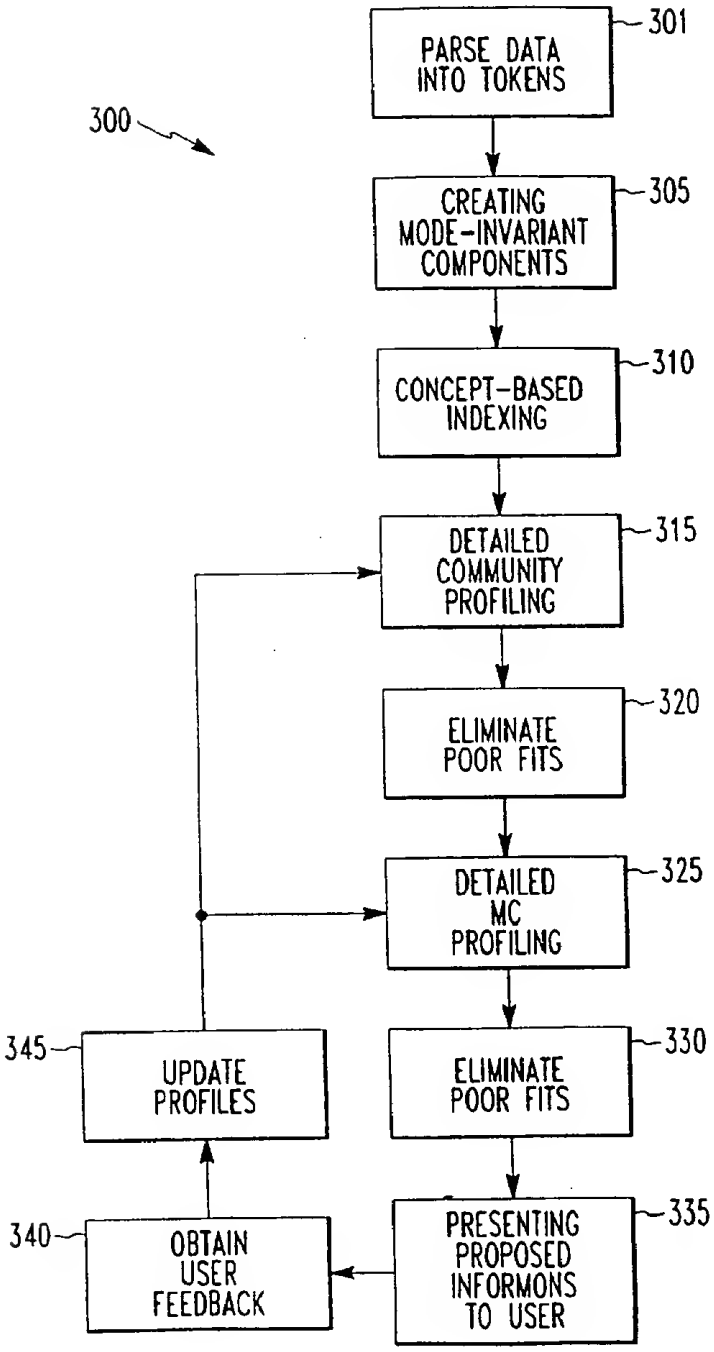

300
PARSE DATA INTO TOKENS
301
CREATING MODE-INVARIANT COMPONENTS
305
CONCEPT-BASED INDEXING
310
DETAILED COMMUNITY PROFILING
315
ELIMINATE POOR FITS
320
DETAILED MC PROFILING
325
ELIMINATE POOR FITS
330
PRESENTING PROPOSED INFORMONS TO USER
335
OBTAIN USER FEEDBACK
340
UPDATE PROFILES
345

FIG. 5

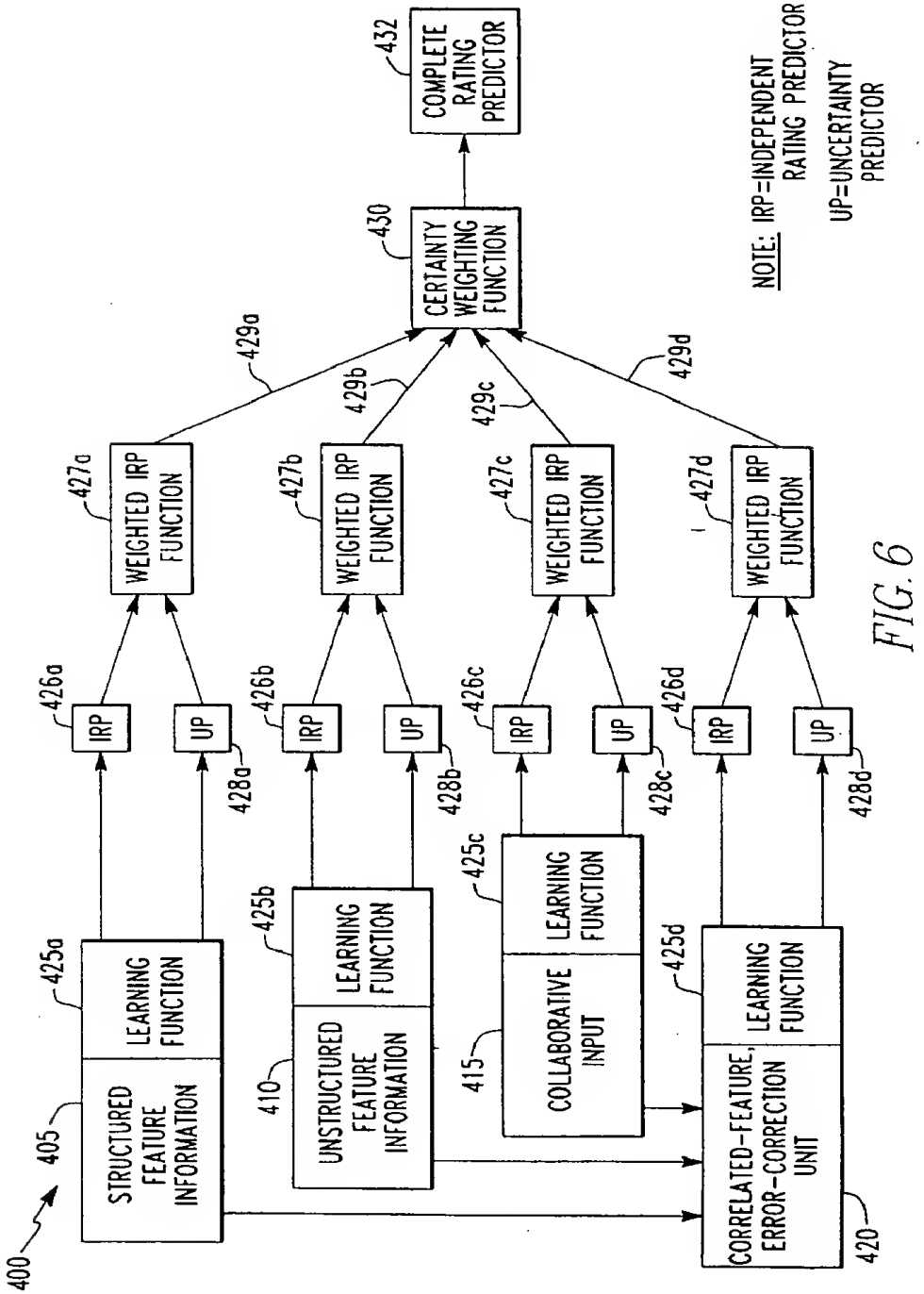

400
405
STRUCTURED FEATURE INFORMATION
410
UNSTRUCTURED FEATURE INFORMATION
415
COLLABORATIVE INPUT
420
CORRELATED-FEATURE, ERROR-CORRECTION UNIT
425a
LEARNING FUNCTION
425b
LEARNING FUNCTION
425c
LEARNING FUNCTION
425d
LEARNING FUNCTION
426a
IRP
426b
IRP
426c
IRP
426d
IRP
428a
UP
428b
UP
428c
UP
428d
UP
427a
WEIGHTED IRP FUNCTION
427b
WEIGHTED IRP FUNCTION
427c
WEIGHTED IRP FUNCTION
427d
WEIGHTED IRP FUNCTION
429a
429b
429c
429d
430
CERTAINTY WEIGHTING FUNCTION
432
COMPLETE RATING PREDICTOR
NOTE: IRP=INDEPENDENT RATING PREDICTOR
UP=UNCERTAINTY PREDICTOR

FIG. 6

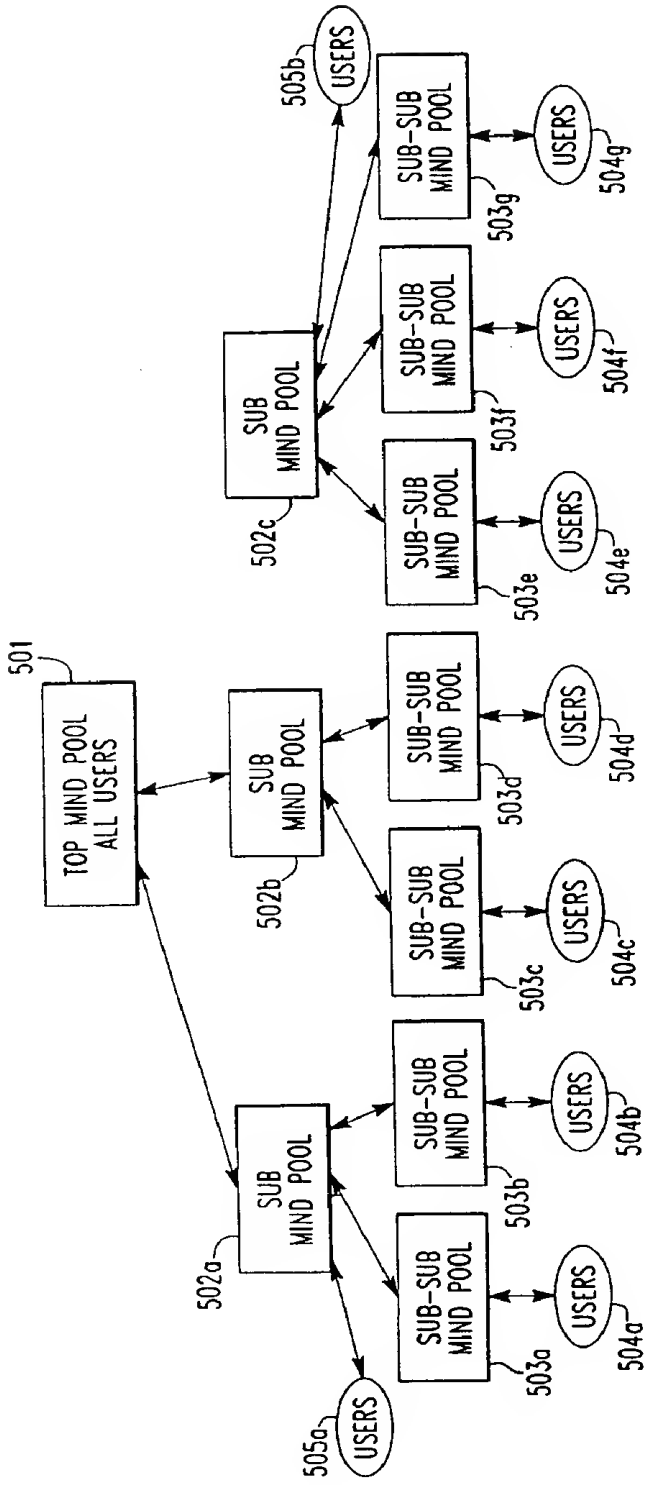

501
TOP MIND POOL ALL USERS
502a
SUB MIND POOL
502b
SUB MIND POOL
502c
SUB MIND POOL
503a
SUB-SUB MIND POOL
503b
SUB-SUB MIND POOL
503c
SUB-SUB MIND POOL
503d
SUB-SUB MIND POOL
503e
SUB-SUB MIND POOL
503f
SUB-SUB MIND POOL
503g
SUB-SUB MIND POOL
504a
USERS
504b
USERS
504c
USERS
504d
USERS
504e
USERS
504f
USERS
504g
USERS
505a
USERS
505b
USERS

FIG. 7

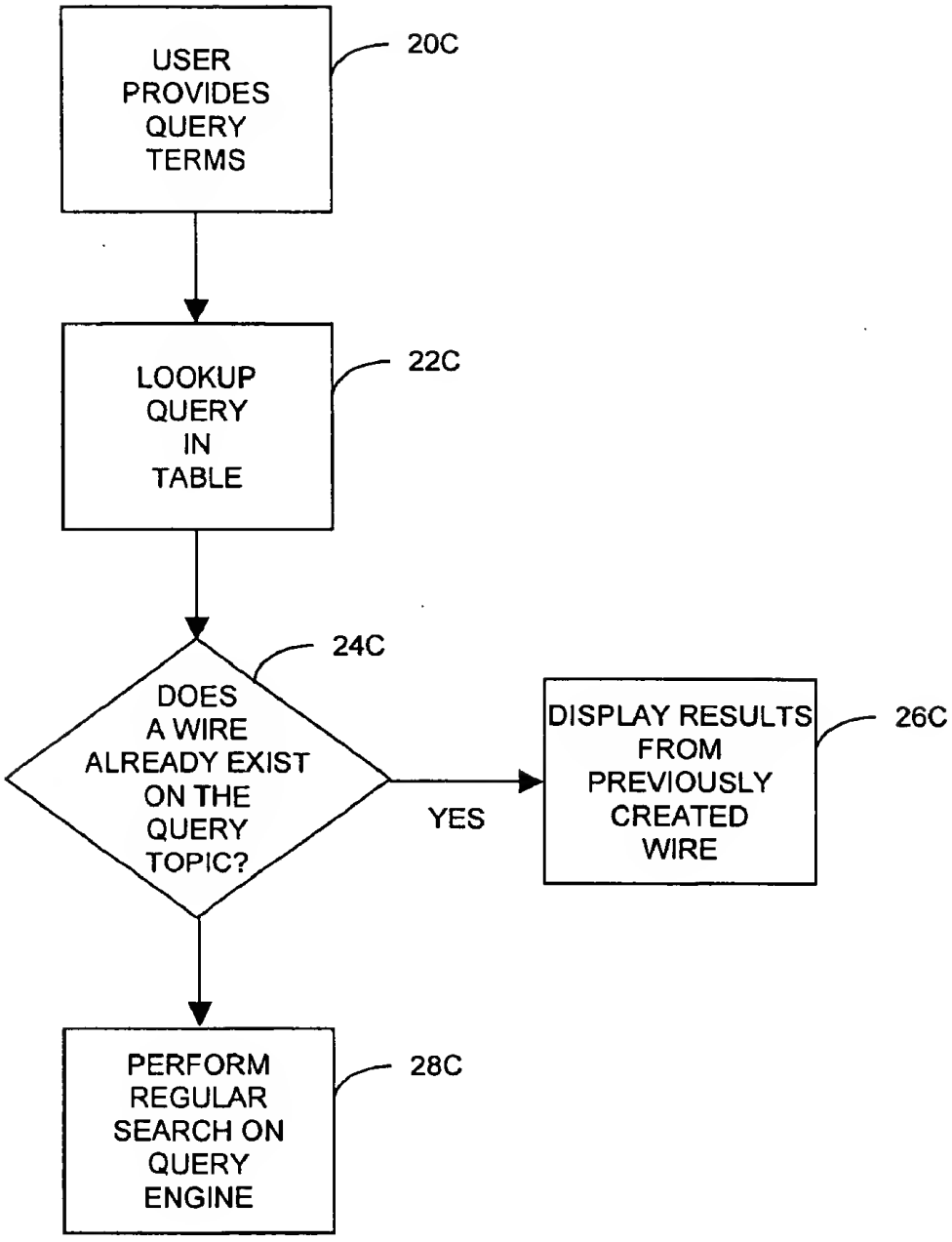

USER PROVIDES QUERY TERMS
20C
LOOKUP QUERY IN TABLE
22C
DOES A WIRE ALREADY EXIST ON THE QUERY TOPIC?
24C
YES
DISPLAY RESULTS FROM PREVIOUSLY CREATED WIRE
26C
PERFORM REGULAR SEARCH ON QUERY ENGINE
28C

FIG. 8

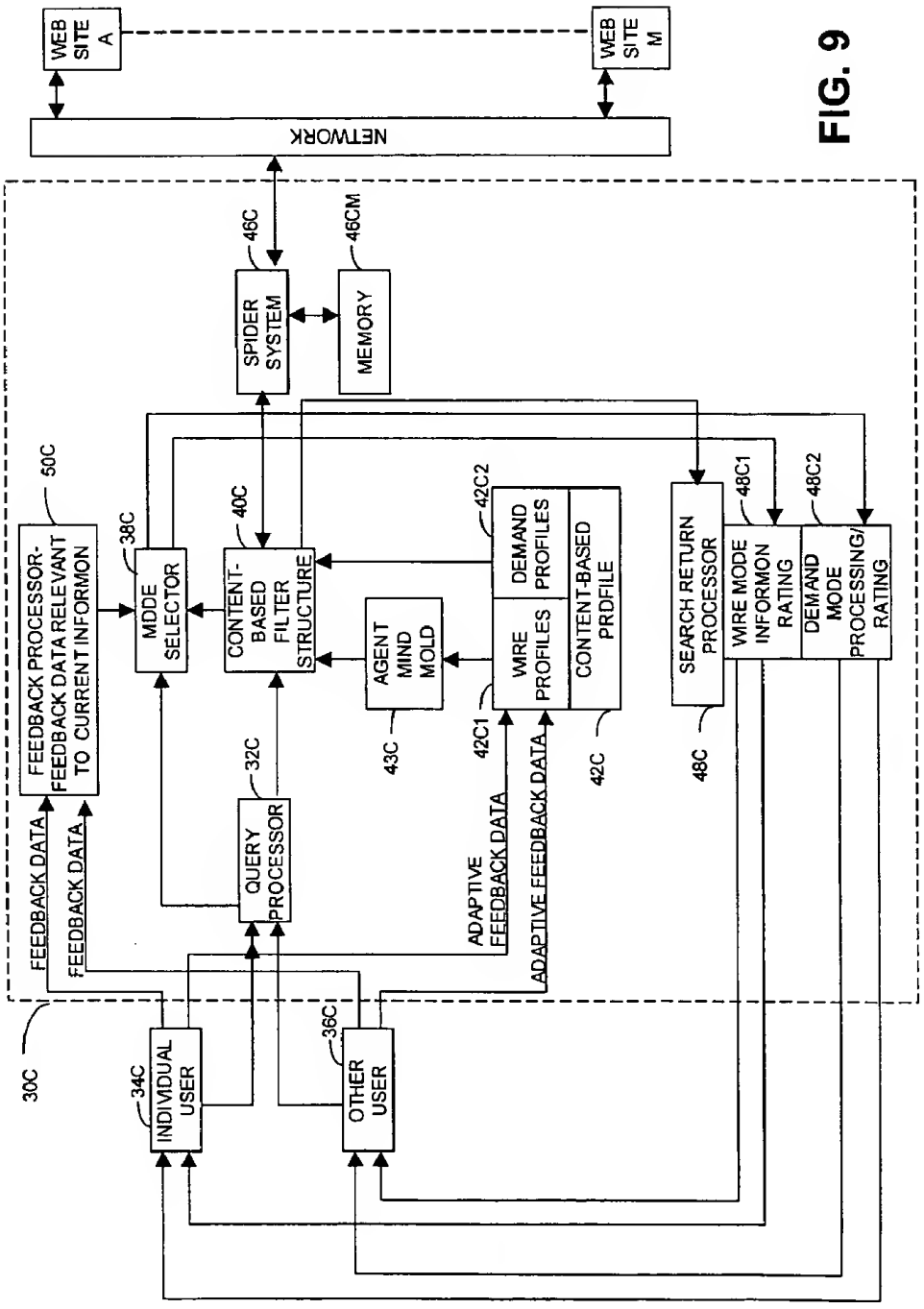

WEB SITE A
WEB SITE M
NETWORK
46C
46CM
SPIDER SYSTEM
MEMORY
50C
FEEDBACK PROCESSOR-FEEDBACK DATA RELEVANT TO CURRENT INFORMON
38C
MDDE SELECTOR
40C
CONTENT-BASED FILTER STRUCTURE
42C2
DEMAND PROFILES
CONTENT-BASED PROFILE
AGENT MIND MOLD
43C
WIRE PROFILES
42C1
42C
SEARCH RETURN PROCESSOR
48C
48C1
WIRE MODE INFORMON RATING
48C2
DEMAND MODE PROCESSING/RATING
32C
QUERY PROCESSOR
FEEDBACK DATA
FEEDBACK DATA
ADAPTIVE FEEDBACK DATA
ADAPTIVE FEEDBACK DATA
30C
34C
INDIVIDUAL USER
36C
OTHER USER

FIG. 9

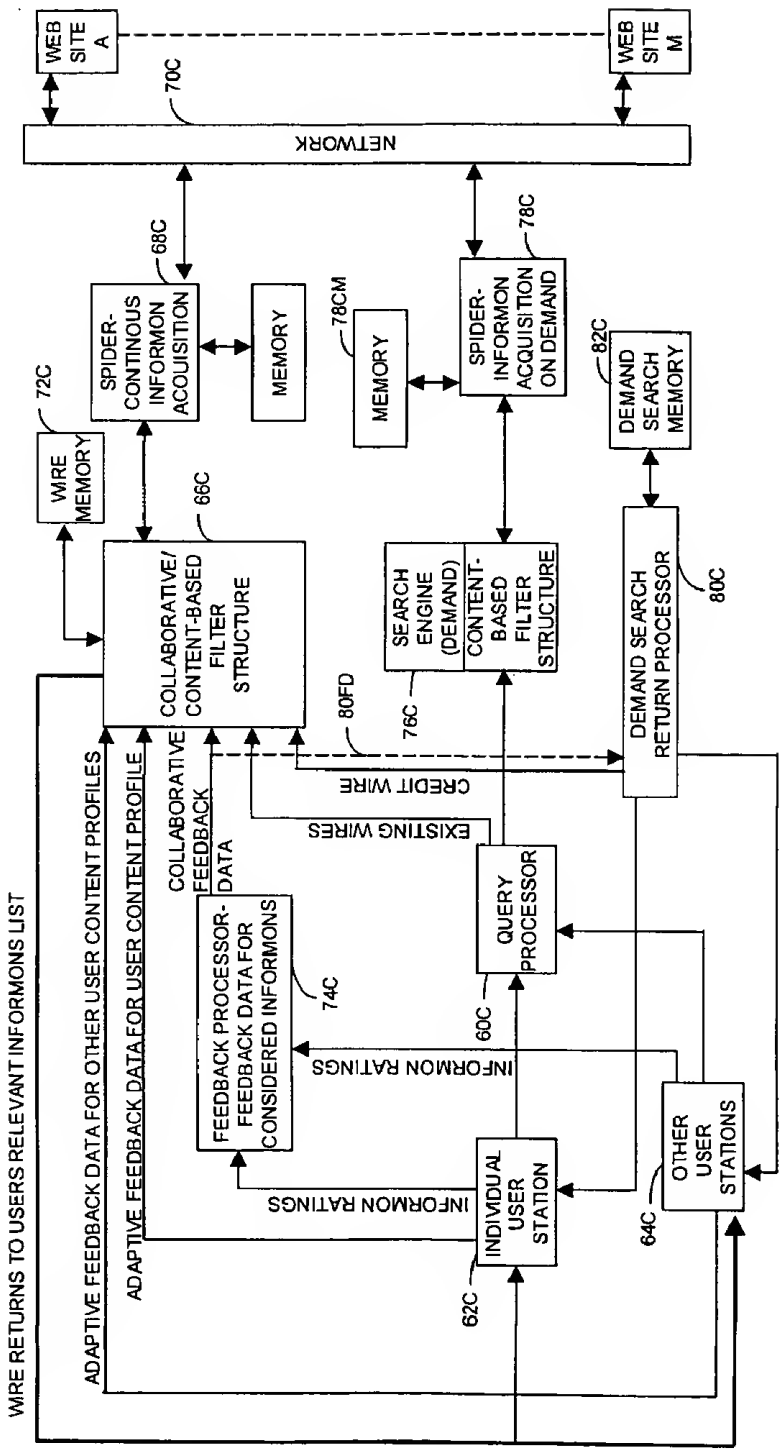

WEB SITE A
WEB SITE M
70C
NETWORK
68C
SPIDER-CONTINOUS INFORMON ACQUISITION
MEMORY
78CM
MEMORY
78C
SPIDER-INFORMON ACQUISITION ON DEMAND
82C
DEMAND SEARCH MEMORY
72C
WIRE MEMORY
66C
COLLABORATIVE/ CONTENT-BASED FILTER STRUCTURE
SEARCH ENGINE (DEMAND)
CONTENT-BASED FILTER STRUCTURE
76C
80FD
DEMAND SEARCH RETURN PROCESSOR
80C
CREDIT WIRE
EXISTING WIRES
COLLABORATIVE FEEDBACK DATA
QUERY PROCESSOR
60C
FEEDBACK PROCESSOR-FEEDBACK DATA FOR CONSIDERED INFORMONS
74C
INFORMON RATINGS
INDIVIDUAL USER STATION
62C
OTHER USER STATIONS
64C
INFORMON RATINGS
ADAPTIVE FEEDBACK DATA FOR OTHER USER CONTENT PROFILES
ADAPTIVE FEEDBACK DATA FOR USER CONTENT PROFILE
WIRE RETURNS TO USERS RELEVANT INFORMONS LIST

FIG. 10

COLLABORATIVE/ADAPTIVE SEARCH ENGINE

This application is a continuation-in-part of copending application Scr. No. 08/627,436 filed on Apr. 4, 1996 now U.S Pat. No. 5,867,799, the entire contents of wbicb are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

The present invention relates to information processing systems for large or massive information networks, such as thc internct, and morc particularly to such information systems especially adapted for operation in portal and other web sites wherein a search engine operates with collaborative and content-based filtering to provide better searcb responses to user queries.

In the operation of the internet, a countless number of information are available for downloading from any of at least thousands of sites for consideration hy a user at the user's location. A user typically connects to a portal or other web site having a search capability, and thereafter enters a particular query, i.e., a request for information relevant to a topic, a field of interest, etc. Thereafter, the search site typically employs a "spider" scanning system and a content-based filter in a search engine to search the internet and find information which match thc query. This process is basically a pre-search process in which matching informons are found, at the time of initiating a search for the user's query, by comparing informons in an "informon data base" to the user's query. In essence, the pre-search process is a short tcrm search for quickly finding and quickly identifying information entities which are content matcbed to tbe user's query.

The return list of matching informons can be very extensive according to thc subject of thc qucry and thc brcadth of the query. More specific queries typically result in shorter return lists. In some cases, the search site may also be structured to find web sites which probably have stored informons matching the entered query.

Collaborative data can be made available to assist in informon rating when a user actually downloads an informon, considers and evaluates it, and returns data to the search site as a representation of the value of the considered informon to the user.

In the patent application which is parent to this continuation-in-part application, i.e. Ser. No. 08/627,436, filed by the present inventors on Apr. 4, 1996, now U.S. Pat. No. 5,867,799 and hereby incorporated by reference, an advanced collaborative/content-based information filter system is employed to provide superior filtering in the process of finding and rating informons which match a user's query. The information filter structure in this system integrates content-based filtering and collaborative filtering to determine relevancy of informons received from various sites in the Internet or other network. In operation, a user enters a query and a corresponding "wire" is cstablishcd, i.e., the query is profiled in storage on a content basis and adaptively updated over time, and informons obtained from the network are compared to the profile for relevancy and ranking. A continuously operating "spider" scans the network to find informons which are received and processcd to determine relevancy to the individual user's wire or to wires established by numerous other users.

The integrated filter system compares received informons to thc individual user's query profilc data, combincd with collaborative data, and ranks, in order of value, informons found to be relevant. The system maintains the ranked informons in a stored list from whicb the individual user can select any listed informon for consideration.

As the system continues to feed the individual user's "wire", the stored relevant informon list typically changes due to factors including a return of new and more relevant informons, adjustments in the user's query, feedback evaluations hy the user for considered informons, and updatings in collaborativc feedback data. Received informons are similarly processed for other users' wires established in the information filter system. Thus, the integrated information filter system performs continued long-term searching, i.e., it compares network informons to multiple users' queries to find matching informons for various users' wires over the course of time, whereas conventional search engines initiate a search in response to an individual user's query and use content-based filtering to compare the query to accessed network informons typically to find matching informons during a limited, short-term search time period.

The present invention is directed to an information processing system especially adapted for use at internet portal or other web sites to make network searches for information entities relevant to user queries, with collaborative feedback data and content-based data and adaptive filter structuring, being used in filtering operations to produce significantly improved search results.

SUMMARY OF THE INVENTION

A search engine system employs a content-based filtering system for receiving informons from a network on a continuing basis and for filtering the informons for relevancy to a wire or demand query from an individual user. A feedback system provides feedback data from other users.

Another system controls the operation of the filtering system to filter for one of a wire response and a demand response and to return the one response to the user. The filtering system combines pertaining feedback data from the feedback system with content profile data in determining the relevancy of the informons for inclusion in at least a wire response to tbe qucry.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is an diagrammatic representation of an embodiment of an information filtering apparatus according to the present invention.

FIG. **2** is an diagrammatic representation of another embodiment of an information filtering apparatus according to the present invention.

FIG. **3** is a flow diagram for an embodiment of an information filtering metbod according to the present invention.

FIG. **4** is a flow diagram for another embodiment of an information filtering method according to the present invention.

FIG. **5** is a flow diagram for yet another embodiment of an information filtering method according to the present invention.

FIG. **6** is an illustration of a three-component-input model and profile with associated predictors.

FIG. **7** is an illustration of a mind pool bierarcby.

FIG. **8** is a logic diagram illustrating a search selection feature of the invention;

FIG. **9** is a functional block diagram of an embodiment of the invention in which an integrated information processing

system employs a search engine and operates with combined collaborative filtering and content-based filtering, which is preferably adaptive, to develop responses to user queries.

FIG. 10 shows another and presently preferred embodiment of the invention in which an information processing system includes an integrated filter structure providing collaborative/adaptive-content-based filtering to develop longer term, continuing responses to user queries, and a search engine structure which provides short term, demand responses to user queries, with the system directing user queries to the appropriate structure for responses.

DETAILED DESCRIPTION OF THE EMBODIMENTS

The invention herein is preferably configured with an apparatus and method for information filtering in a computer system receiving a data stream from a computer network, in which entities of information relevant to the user, or "informons," are extracted from the data stream using content-based and collaborative filtering. The information filtering is long term in the sense that it operates on a continuing basis, and is both interactive and distributed in structure and method. It is interactive in that communication is substantially big-directional at each level of the filter. It is distributed in that all or part of the information filter can include a purely hierarchical (up-and-down/parent-child) structure or method, a purely parallel (peer-to-peer) structure or method, or a combination of hierachical and parallel structures and method.

As used herein, the term "informon" comprehends an information entity of potential or actual interest to a particular user. In general, informons can be heterogenous in nature and can be all or part of a textual, a visual, or an audio entity. Also, informons can be composed of a combination of the aforementioned entities, thereby being a multimedia entity. Furthermore, an informon can be an entity of patterned data, such as, a data file containing a digital representation of signals and can be a combination of any of the previously-mentioned entities. Although some of the data in a data stream, including informons, may be included in an informon, not all data is relevant to a user, and is not within the definition of an informon. By analogy, an informon may be considered to be a "signal," and the total data stream may be considered to be "signal +noise." Therefore, an information filtering apparatus is analogous to other types of signal filters in that it is designed to separate the "signal" from the "noise."

Also as used herein, the term "user" is an individual in communication with the network. Because an individual user can be interested in multiple categories of information, the user can be considered to be multiple clients each having a unique profile, or set of attributes. Each member client profile, then, is representative of a particular group of user preferences. Collectively, the member client profiles associated with each user is the user profile. The present invention can apply the learned knowledge of one of a user's member clients to others of the user's member clients, so that the importance of the learned knowledge, e.g., the user's preference for a particular author in one interest area as represented by the member client, can increase the importance of that particular factor, A's authorship, for others of the user's member clients. Each of the clients of one user can be associated with the individual clients of other users insofar as the profiles of the respective clients have similar attributes. A "community" is a group of clients, called member clients, that have similar member client profiles, i.e., that share a subset of attributes or interests. In general, the subset of shared attributes forms the community profile for a given community and is representative of the community norms, or common client attributes.

The "relevance" of a particular informon broadly describes how well it satisfies the user's information need. The more relevant an informon is to a user, the higher the "signal" content. The less relevant the informon, the higher the "noise" content. Clearly, the notion of what is relevant to a particular user can vary over time and with context, and the user can find the relevance of a particular informon limited to only a few of the user's potentially vast interest areas. Because a user's interests typically change slowly, relative to the data stream, it is preferred to use adaptive procedures to track the user's current interests and follow them over time. Provision, too, is preferred to be made for sudden changes in interest, e.g., taking up antiquarian sword collecting and discontinuing stamp collecting, so that the method and apparatus track the evolution of "relevance" to a user and the communities of which the user is a member. In general, information filtering is the process of selecting the information that a users wishes to see, i.e., informons, from a large amount of data. Content-based filtering is a process of filtering by extracting features from the informon, e.g., the text of a document, to determine the informon's relevance. Collaborative filtering, on the other hand, is the process of filtering informons, e.g., documents, by determining what informons other users with similar interests or needs found to be relevant.

The system apparatus includes a filter structure having adaptive content based filters and adaptive collaborative filters, which respectively include, and respond to, an adaptive content profile and an adaptive collaboration profile. As used herein, the term "content-based filter" means a filter in which content data, such as key words, is used in performing the filtering process. In a collaborative filter, other user data is used in performing the filtering process. A collaborative filter is also sometimes referred to as a "content" filter since it ultimately performs the task of finding an object or document having content relevant to the content desired by a user. If there are some instances herein where the term "content filter" is used as distinguished from a collaborative filter, it is intended that the term "content filter" mean "content-based filter." The adaptive filters each are preferred to include at least a portion of a community filter for each community serviced by the apparatus, and a portion of a member client filter for each member client of the serviced communities. For this reason, the adaptive filtering is distributed in that each of the community filters perform adaptive collaborative filtering and adaptive content filtering, even if on different levels, and even if many filters exist on a given level. The integrated filtering permits an individual user to be a unique member client of multiple communities, with each community including multiple member clients sharing similar interests. The adaptive features permit the interests of member clients and entire communities to change gradually over time. Also a member client has the ability to indicate a sudden change in preference, e.g., the member client remains a collector but is no longer interested in coin collecting.

The filter structure also implements adaptive credibility filtering, providing member clients with a measure of informon credibility, as judged by other member clients in the community. For example, a new member client in a first community, having no credibility, can inject an informon into the data flow, thereby providing other member clients in other communities with the proposed informon, based on the

respective community profile and member client profiles. If the other member clients believe the content of the informon to be credible, the adaptive credibility profile will reflect a growing credibility. Conversely, feedback profiles from informon recipients that indicate a lack of credibility cause the adaptive credibility profile, for the informon author to reflect untrustworthiness. However, the growth and declination of credibility are not "purely democratic," in the sense that one's credibility is susceptible to the bias of others' perceptions, so the growth or declination of one's credibility is generally proportional to how the credibility of the new member client is viewed by other member clients.

Member clients can put their respective reputations "on the line," and engage in spirited discussions which can be refereed hy other interested member clients. The credibility profile further can be partitioned to permit separate credibility sub-profiles for the credibility of the content of the informon, the author, the author's community, the reviewers, and the like, and can be fed back to discussion participants, reviewers, and observers to monitor the responses of others to the debate. The adaptive credibility profiles for those member clients with top credibility ratings in their communities may be used to establish those member clients as "experts" in their respective communities.

With this functionality, additional features can bc implemented, including, for example, "instant polling" on a matter of political or consumer interest. In conjunction with both content and collaborative filtering, credibility filtering, and the resulting adaptive credibility profiles, also may be used to produce other features, such as on-line consultation and recommendation services. Although the "experts" in the communities most closely related to the topic can be afforded special status as such, member clients from other communities also can participate in the consultation or recommendation process.

In one embodiment of the consultation servicc, credibility filtering can he augmented to include consultation filtering. With this feature, a member client can transmit an informon to the network with a request for guidance on an issue, for example, caring for a sick tropical fish. Other member clients can rcspond to thc rcquester with informons related to the topic, e.g., suggestions for water temperature and antihiotics. The informons of the responders can include their respective credibility profiles, community membership, and professional or avocational affiliations. The requester can provide feedback to each of the responders, including a rating of the credibility of the responder on the particular topic. Additionally, the responders can accrue quality points, value tokens, or "info hucks," as apportioned by the requester, in return for useful guidance.

Similarly, one embodiment of an on-line recommendation service uses recommendation filtering and adaptive recommendation profiles to give member clients recommendations on matters as diverse as local auto mechanics and world-class medieval armor refurbisbers. In this embodiment, the requester can transmit the informon to the network bearing tbe request for recommendation. Other member clients can respond to tbe requester witb informons having specific recommendations or dis-recommendations, advice, etc. As with the consultation service, the informons of the responders can be augmented to include their respective credibility profiles, community membership, and professional or avocational affiliations. A rating of each recommendation provided by a responder, relative to other responders' recommendations, also can be supplied. The requester can provide fcedback to each of the rcsponders, including a rating of the credibility of the responder on the particular topic, or the quality of the recommendation. As before, the responders can accrue quality points, value tokens, or "info bucks," as apportioned by the requester, in return for the useful recommendation.

Furthermore, certain embodiments are preferred to be self-optimizing in that some or all of the adaptive filters used in the system dynamically seek optimal values for the function intended by the filter, e.g., content analysis, collaboration, credibility, reliability, etc.

The filter structure herein is capable of identifying, the preferences of individual member clients and communities, providing direct and inferential consumer preference information, and tracking shifts in the preferences whetber the shifts be gradual or sudden. The consumer preference information can be used to target particular consumer preference groups, or cohorts, and provide members of the cohort with targeted informons relevant to their consumer preferences. This information also may be used to follow demographical shifts so that activities relying on accurate demographical data, such as retail marketing, can use tbe consumer preference information to anticipate evolving consumer needs in a timcly manner.

To provide a basis for adaptation, it is preferred that each raw informon be processed into a standardized vector, which may be on the order of 20,000 to 100,000 tokens long. The learning and optimization methods that ultimately are chosen are preferred to be suhstantially robust to the problems which can be presented by such high-dimensional input spaces. Dimensionality reduction using methods such as the singular value decomposition (SVD), or auto-encoding neural networks attempt to reduce the size of the space while initially retaining the information contained in the original representation. However, the SVD can lose information during the transformation and may give inferior results. Two adaptation/learning methods that are presently preferred include the TF-IDF technique and the MDL technique.

FIG. **1** illustrates one embodiment of an information filtering apparatus **1** structured for search engine implementation in accordance with the invention as described subsequently herein in connection with FIGS. **8** and **9**. In general, a data stream is conveyed through network **3**, which can be a global internet work. A skilled artisan would recognize that apparatus **1** can bc used with othcr types of networks, including, for example, an enterprise-wide network, or "intranet." Using network 3, User #1 (**5**) can communicate with other users, for example, User #2 (**7**) and User #3 (**9**), and also with distributed network resources such as resource #1 (**11**) and resource #2 (**13**).

Apparatus **1** is preferred to be part of computer system **16**, although User #1 (**5**) is not required to be the sole user of computer system **16**. In one present embodiment, it is preferred that computer system **16** having information filter apparatus **1** therein filters information for a plurality of users. One application for apparatus **1**, for example, could he that user **5** and similar users may be subscribers to a commercial information filtering service, which can be provided by the owner of computer system **16**.

Extraction means **17** can be coupled with, and receives data stream **15** from, network **3**. Extraction means **17** can identify and extract raw informons **19** from data stream **15**.

Each of the raw informons **19** has an information content. Extraction means **17** uses the adaptive content filter, and at least part of the adaptive content profile, to analyze the data stream for the presence of raw informons. Raw informons are those data entities whose content idcntifies them as being "in the ballpark," or of potential interest to a community

coupled to apparatus **1**. Extraction means **17** can remove duplicate informons, even if tbe informons arrive from different sources, so that user resources are not wasted by handling and viewing rcpctitivc and cumulative information. Extraction means **17** also can use at least part of a community profile and a user profile for User #1 (**5**) to determine whetber tbe informon content is relevant to the community of which User #1 is a part.

Filter means **21** adaptively filters raw informons **19** and produces proposed informons **23** which are conveyed to User #1 (**5**) by communication means **25**. A proposed informon is a selected raw informon that, based upon tbe respective member client and community profiles, is predicted to be of particular interest to a member client of User **5**. Filter means **21** can include a plurality of community filters **27***a,b* and a plurality of member client filters **28***a–e*, eacb respectively having community and member client profiles. Wben raw informons **19** are filtered by filter means **21**, those informons that are predicted to be suitable for a particular member client of a particular community, e.g., User #1 (**5**), responsive to tbc respective community and member client profiles, are conveyed thereto. Where such is desired, filter means **21** also can include a credibility filter **35** wbicb enables means **21** to perform credibility filtering of raw informons **19** according to a credibility profile.

It is preferred that the adaptive filtering performed within filter means **21** by the plurality of filters **27***a,b*, **28***a–e*, and **35**, use a self-optimizing adaptive filtering so that each of the pararoeters processed by filters **27***a,b*, **28***a–e*, and **35**, is driven continually to respective values corresponding to a roinimal error for eacb individual parameter. Self-optimization encourages a dynamic, marketplace-like operation of the system, in tbat tbose entities having the most desirable value, e.g., highest credibility, lowest predicted error, etc., are favored to prevail.

Self-optimization can be effected according to respective preselected self-optimizing adaptation techniques including, for example, one or more of a top-key-word-selection adaptation technique, a nearest-neigbbor adaptation technique, a term-weigbting adaptation technique, a probabilistic adaptation technique, and a neural network learning technique. In one present embodiment of tbe invention, tbe terro-weigbting adaptation tecbnique is preferred to be a TF-IDF technique and the probabilistie adaptation technique is preferred to be a MDL technique.

Wben user **5** receives proposed informon **23** from apparatus **1**, user **5** is provided with multiple feedback queries along with thc proposed informon. By answering, user **5** creates a feedback profile that corresponds to feedback response **29**. User feedback response **29** can be active feedback, passive feedback, or a combination. Active feedback can include the user's numerical rating for an informon, hints, and indices. Hints can include like or dislike of an author, and informon source and timeliness. Indices can include credibility, agreement with consent or author, bumor, or value. Feedback response **29** provides an actual response to proposed informon **23**, which is a measure of tbe relevance of the proposed informon to tbe information need of user **5**. Sucb relevance feedback attempts to improve tbe performance for a particular profile by modifying thc profiles, based on feedback response **29**.

'A predicted response anticipated by adaptive filtering means **21** can be compared to the actual feedback response **29** of user **5** by first adaptation means **30**, which derives a prediction error. First adaptation means **30** also can include prediction means **33**, wbich collects a number of temporally-spaced feedback responses, to update tbe adaptive collaboration profile, tbe adaptive content profile, or both, witb an adapted future prediction **34**, in order to minimize subsequent prediction errors by tbe respeetive adaptive collaboration filter and adaptive content filter.

In one embodiment of tbe invention herein, it is preferred tbat prediction means **33** be a self-optimizing prediction means using a preselected learning technique.

Such techniques can include, for example, one or more of a op-key-word-selection learning tecbnique, a nearest-neighbor learning technique, a term-weigbting learoing technique, and a probabilistic learning tecbnique. First adaptation means **30** also can include a neural network tberein and eroploy a neural network learning technique for adaptation and prediction. In one present embodiment of tbc invention, tbe term-weighting learning tecbnique is preferred to be a TF-IDF technique and the probabilistic learning tecbnique is preferred to be a MDL learoing tecbnique.

First adaptation means **30** furtber can include second adaptation means **32** for adapting at least one of the adaptive collaboration profiles, the adaptive content profiles, tbe community profile, and the user profile, responsive to at least one of the other profiles. In this manner, trends attributable to individual member clients, individual users, and individual communities in one domain of system **16** can be recognized by, and influence, similar entities in other domains (melding agent "minds"), contained within system **16** to the extent tbat the respective entities share common attributes.

Apparatus **1** also can include a computer storage means **31** for storing the profiles, including the adaptive contcnt profile and the adaptive collaboration profile. Additional trend-tracking information can be stored for later retrieval in storage means **31**, or may be conveyed to network **3** for remote analysis, for example, by User #2 (**7**).

FIG. **2** illustrates anotber preferred embodiment of information filtering apparatus **50**, in computer system **51**. Apparatus **50** can include first processor **52**, second processor **53***a,b*, tbird processor **64***a–d*, and a fourth processor **55**, to effect thc desired information filtering. First processor **52** can be coupled to, and receive a data stream **56** from, network **57**. First processor **52** can serve as a pre-processor by extracting raw informons **58** from data stream **56** responsive to preprocessing profile **49** and conveying informons **58** to second processor **53***a,b*.

Because of tbe inconsistencies presented by the nearly-infinite individual differences in the modes of conceptualization, expression, and vocabulary among users, even within a community of coinciding interests, similar notions can be described witb vastly different terms and connotations, greatly complicating informon characterization. Mode variations can be even greater betweon disparate communities, discouraging interaction and knowledge-sharing among communities. Therefore, it is particularly preferred tbat processor **52** create a mode-invariant representation for each raw informon, thus allowing fast, accurate informon cbaracterization and collaborative filtering. Mode-invariant representations tend to facilitate relevant informon selection and distribution within and among communities, thereby promoting knowledge-sharing, thereby benefitting tbe group of interlinked communities, i.e., a society, as well.

First processor **52** also can be used to prevent duplicate informons, e.g., tbe same information from different sources, from further penetrating, and thus consuming the resources of, thc filtering process. Othcr processors **53***,a,b*, **54***a–d*, also may be used to perform tbe duplicate informa-

tion elimination function, but additionally may measure the differences between the existing informon and new informons. That difference between the content of the informon the previous time the user reviewed it and the content of the informon in its present form is the "delta" of interest. Processors **53***a,b*, **54***a–d* may eliminate the informon from further processing, or direct the new, altered informon to the member client, in the event that nature or extent of the change exceeds a "delta" threshold. In general, from the notion of exceeding a preselected delta threshold, one may infer that the informon has changed to the extent that the change is interesting to the user. The nature of this change can be shared among all of a user's member clients. This delta threshold can he preselected hy the user, or by the preselected learning technique. Such processing, or "delta learning" can he accomplished by second processor **53***a,b*, alone or in concert with third processor **54***a–d*. Indeed, third processor **54***a–d* can be the locus for delta learning, where processor **54***a–d* adapts a delta learning profile for each member client of the community, i.e. user, thus anticipating those changes in existing informons that the user may find "interesting."

Second processor **53***a,b* can filter raw informons **58** and extract proposed community informons **59***a,b* therefrom. Informons **59***a,b* are those predicted by processor **53***a,b* to be relevant to the respective communities, in response to a community profiles **48***a,b* that are unique to the communities. Although only two second processors **53***a,b* are shown in FIG. **2**, system **51** can be scaled to support many more processors, and communities. It is presently preferred that second processor **53***a,b* extract community informons **59***a,b* using a two-step process. Where processor **52** has generated mode-invariant concept representations of the raw informons, processor **53***a,b* can perform concept-based indexing, and then provide detailed community filtering of each informon.

Third processors **54***a–d* can receive community informons **59***a,b* from processors **53***a,b*, and extract proposed member client informons **61***a–d* therefrom, responsive to unique member client profiles **62***a–d* for respective ones of member clients **63***a–d*. Each user can be represented by multiple member clients in multiple communities. For example, each of users **64***a,b* can maintain interests in each of the communities serviced by respective second processors **53***a,b*, and each receive separate member client informons **61***b,c* and **61***a,d*, respectively.

Each member client **63***a–d* provides respective member client feedback **65***a–d* to fourth processor **55**, responsive to the proposed member client informons **61**a–d. Based upon the member client feedback **65***a–d*, processor **55** updates at least one of the preprocessing profile **49**, community profiles **48***a,b* and member client profiles **62***a–d*. Also, processor **55** adapts at least one of the adaptive content profile **68** and the adaptive collaboration profile **69**, responsive to profiles **49**, **48***a,b*, and **62***a–d*.

Fourth processor **55** can include a plurality of adaptive filters **66***a–d* for each of the aforementioned profiles and computer storage therefor. It is preferred that the plurality of adaptive filters **66***a–d* be self-optimizing adaptive filters. Self-optimization can be effected according to a preselected self-optimizing adaptation technique including, for example, one or more of a top-key-word-selection adaptation technique, a nearest-neighbor adaptation technique, a term-weighting adaptation technique, and a probabilistic adaptation technique. Any of the adaptive filters **66***a–d* may include a neural network In one present embodiment of the invention, the term-weighting adaptation technique is preferred to be a TF-IDF technique and the probabilistic adaptation technique is preferred to be a MDL technique.

An artisan would recognize that one or more of the processors **52–55** could be combined functionally so that the actual number of processors used in the apparatus **50** could be less than, or greater than, that illustrated in FIG. **2**. For example, in one embodiment of the present invention, first processor **52** can be in a single microcomputer workstation, with processors **53–55** being implemented in additional respective microcomputer systems. Suitable microcomputer systems can include those based upon the Intel® Pentium-Pro™ microprocessor. In fact, the flexibility of design presented by the invention allows for extensive scalability of apparatus **50**, in which the number of users, and the communities supported may be easily expanded by adding suitable processors. As described in the context of FIG. **1**, the interrelation of the several adaptive profiles and respective filters allow trends attributable to individual member clients, individual users, and individual communities in one domain of system **51** to be recognized hy, and influence, similar entities in other domains, of system **51** to the extent that the respective entities in the different domains share common attributes.

The above described system operates in accordance with **100** for information filtering in a computer system, as illustrated in FIG. **3**, which includes providing a dynamic informon characterization (step **105**) having a plurality of profiles encoded therein, including an adaptive content profile and an adaptive collaboration profile; and adaptively filtering the raw informons (step **110**) responsive to the dynamic informon characterization, thereby producing a proposed informon. The method continues by presenting the proposed informon to the user (step **115**) and receiving a feedback profile from the user (step **120**), responsive to the proposed informon. Also, the method includes adapting at least one of the adaptive content profile (step **125**) and the adaptive collaboration profile responsive to the feedback profile; and updating the dynamic informon characterization (step **130**) responsive thereto.

The adaptive filtering (step **110**) in method **100** can be machine distributed adaptive filtering that includes community filtering (sub-step **135**), using a community profile for each community, and client filtering (sub-step **140**), similarly using a member client profile for each member client of each community. It is preferred that the filtering in sub-steps **135** and **140** be responsive to the adaptive content profile and the adaptive collaboration profile. Method **100** comprehends servicing multiple communities and multiple of users. In turn, each user may be represented by multiple member clients, with each client having a unique member client profile and being a member of a selected community. It is preferred that updating the dynamic informon characterization (step **130**) further include predicting selected subsequent member client responses (step **150**).

Method **100** can also include credibility filtering (step **155**) of the raw informons responsive to an adaptive credibility profile and updating the credibility profile (step **160**) responsive to the user feedback profile. Method **100** further can include creating a consumer profile (step **165**) responsive to the user feedback profile. In general, the consumer profile is representative of predetermined consumer preference criteria relative to the communities of which the user is a member client. Furthermore, grouping selected ones (step **170**) of the users into a preference cohort, responsive to the preselected consumer preference criteria, can facilitate providing a targeted informon (step **175**), such as an advertisement, to the preference cohort.

FIG. 4 illustrates yet another preferred method **200**. In general, method **200** includes partitioning (step **205**) each user into multiple member clients, each having a unique member client profile with multiple client attributes and grouping member clients (step **210**) to form a multiple communities with each member client in a particular community sbaring selected client attributes with other member clients, thereby providing each community with a unique community profile having common client attributes.

Method **200** continues by predicting a community profile (step **215**) for each community using first prediction criteria, and predicting a member client profile (step **220**) for a member client in a particular community using second prediction criteria. Method **200** also includes the steps of extracting raw informons (step **225**) from a data stream and selecting proposed informons (step **230**) from raw informons. The proposed informons generally are correlated with one or more of the common client attributes of a community, and of the member client attributes of the particular member client to whom the proposed informon is offered. After providing the proposed informons to the user (step **235**), receiving user feedback (step **240**) in response to the proposed informons permits the updating of the first and second prediction criteria (step **245**) responsive to the user feedback.

Method **200** further may include prefiltering tbe data stream (step **250**) using the predicted community profile, with the predicted community profile identifying the raw informons in the data stream.

Step **230** of selecting proposed informons can include filtering the raw informons using an adaptive content filter (step **255**) responsive to the informon content; filtering the raw informons using an adaptive collaboration filter (step **260**) responsive to the common client attributes for the pertaining community; and filtering the raw informons using an adaptive member client filter (step **265**) responsive to the unique member client profile.

It is preferred tbat updating the first and second prediction criteria (step **245**) employ a self-optimizing adaptation technique, including, for example, one or more of a top-key-word-selection adaptation technique, a nearest-neighbor adaptation technique, a term-weighting adaptation technique, and a probabilistic adaptation technique. It is further preferred that the term-weighting adaptation technique be a TF-IDF technique and the probabilistic adaptation technique be a minimum description length technique.

The information filtering method shown in FIG. **5** provides rapid, efficient data reduction and routing, or filtering, to the appropriate member client. The method **300** includes parsing the data stream into tokens (step **301**); creating a mode-invariant (MI) profile of the informon (step **305**); selecting the most appropriate communities for each informon, based on the MI profile, using concept-based indexing (step **310**); detailed analysis (step **315**) of each informon with regard to its fit within each community; eliminating poor-fitting informons (step **320**); detailed filtering of each informon relative to fit for each member client (step **325**); eliminating poor-fitting informons (step **330**); presenting the informon to tbe member client/user (step **335**); and obtaining the member client/user response, including multiple ratings for different facets of the user's response to the informon (step **340**).

It is preferred that coherent portions of the data stream, i.e., potential raw informons, be first parsed (step **301**) into generalized words, called tokens. Tokens include punctuation and other specialized symbols that may be part of the structure found in the article headers. For example, in addition to typical words such as "seminar" counting as tokens, the punctuation mark "$" and the symbol "Newsgroup:comp.ai" are also tokens. Using noun phrases as tokens also can be useful.

Next a vector of token counts for the document is created. This vector is the size of the total vocabulary, with zeros for tokens not occurring in the document. Using this type of vector is sometimes called the bag-of-words model. Wbile the bag-of-words model does not capture the order of the tokens in the document, which may be needed for linguistic or syntactic analysis, it captures most of the information needed for filtering purposes.

Although, it is common in information retrieval systems to group tbe tokens together by their common linguistic roots, called stemming, as a next step it is preferred in the present invention that the tokens be left in their unstemmed form. In tbis form, the tokens are amenable to being classified into mode-invariant concept components.

Creating a mode-invariant profile (step **305**), C, includes creating a conceptual representation for each informon, A, that is invariant with respect to the form-of-expression, e.g., vocabulary and conceptualization. Each community can consist of a "Meta-U-Zine" collection, M, of informons. Based upon profile C, the appropriate communities, if any, for each informon in the data stream are selected by concept-based indexing (step **310**) into each M. That is, for each concept C that describes A, put A into a queue Q_M, for each M which is related to C. It is preferred tbat tbere is a list of Ms that is stored for each concept and that can be easily index-searched. Each A that is determined to be a poor fit for a particular M is eliminated from further processing. Once A has been matched with a particular M, a more complex community profile P_M is developed and maintained for each M (step **315**). If A has fallen into Q_M, then A is analyzed to determine wbether it matches P_M strongly enough to be retained or "weeded" out (step **325**) at this stage.

Each A for a particular M is sent to each user's personal agent, or member client U of M, for additional analysis based on tbe member client's profile (step **325**). Each A that fits U's interests sufficiently is selected for U's personal informon, or "U-Zine," collection, Z. Poor-fitting informons are eliminated from placement in Z (step **330**). This user-level stage of analysis and selection may be performed on a centralized server site or on the user's computer.

Next, the proposed informons are presented to user U (step **335**) for review. User U reads and rates each selected A found in Z (step **340**). The feedback from U can consist of a rating for how "interesting" U found A to be, as well as one or more of the following:

- Opinion feedback: Did U agree, disagree, or have no opinion regarding the position of A?
- Credibility Feedback: Did U find the facts, logic, sources, and quotes in A to be truthful and credible or not?
- Informon Qualities: How does the user rate the informons qualities, for example, "interestingness," credibility, funniness, content value, writing quality, violence content, sexual content, profanity level, business importance, scientific merit, surprise/unexpectedness of information content, artistic quality, dramatic appeal, entertainment value, trendiness/importance to future directions, and opinion agreement.
- Specific Reason Feedback: Why did the user like or dislike A?
  - Because of the authority?
  - Because of the source?

Because A is out-of-date (e.g. weather report from 3 weeks ago)?

Because the information contained in A has been seen already? (I.e., the problem of duplicate information delivery)

Categorization Feedback: Did U liked A? Was it placed within the correct M and Z?

Such multi-faceted feedback queries can produce rich feedback profiles from U that can be used to adapt each of the profiles used in the filtering process to some optimal operating point.

One embodiment of creating a MI profile (step **305**) for each concept can include concept profiling, creation, and optimization. Broad descriptors can be used to create a substantially-invariant concept profile, ideally without the word choice used to express concept C. A concept profile can include positive concept clues (PCC) and negative concept clues (NCC). The PCC and NCC can be combined by a processor to create a measure-of-fit that can he compared to a predetermined threshold. If the combined effect of the PCC and NCC exceeds the predetermined threshold, then informon A can be assumed to be related to concept C; otherwise it is eliminated from further processing. PCC is a set of words, phrases, and other features, such as the source or the author, each with an associated weight, that tend to be in A which contains C. In contrast, NCC is a set of words, phrases, and other features, such as the source or the author, each with an associated weight that tend to make it more unlikely that A is contained in C. For example, if the term "car" is in A, then it is likely to be about automobiles. However, if the phrase "bumper car" also is in A, then it is more likely that A related to amusement parks. Therefore, "bumper car" would fall into the profile of negative concept clues for the concept "automobile."

Typically, concept profile C can be created by one or more means. First, C can be explicitly created by user U.

Second, C can be created by an electronic thesaurus or similar device that can catalog and select from a set of concepts and the words that can be associated with that concept. Third, C can be created by using co-occurrence information that can be generated by analyzing the content of an informon. This means uses the fact that related features of a concept tend to occur more often within the same document than in general. Fourth, C can be created by the analysis of collections, H, of A that have been rated by one or more U. Combinations of features that tend to occur repeatedly in H can be grouped together as PCC for the analysis of a new concept. Also, an A that one or more U have rated and determined not to be within a particular Z can be used for the extraction of NCC.

Concept profiles can be optimized or learned continually after their creation, with the objective that nearly all As that Us have found interesting, and belonging in M, should pass the predetermined threshold of at least one C that can serve as an index into M. Another objective of concept profile management is that, for each A that does not fall into any of the one or more M that are indexed by C, the breadth of C is adjusted to preserve the first objective, insofar as possible. For example, if C's threshold is exceeded for a given A, C's breadth can be narrowed by reducing PCC, increasing NCC, or both, or by increasing the threshold for C.

In the next stage of filtering, one embodiment of content-based indexing takes an A that has been processed into the set of C that describe it, and determine which M should accept the article for subsequent filtering, for example, detailed indexing of incoming A. It is preferred that a data structure including a database be used, so that the vector of Ms, that are related to any concept C, may be looked-up. Furthermore, when a Z is created by U, the concept clues given by U to the information filter can be used to determine a set of likely concepts C that describe what U is seeking. For example, if U types in "basketball" as a likely word in the associated Z, then all concepts that have a high positive weight for the word "basketball" are associated with the new z. If no such concepts C seem to pre-exist, an entirely new concept C is created that is endowed with the clues U has given as the starting profile.

To augment the effectiveness of concept-based indexing, it is preferred to provide continual optimization learning. In general, when a concept C no longer uniquely triggers any documents that have been classified and liked by member clients U in a particular community M, then that M is removed from the list of M indexed into by C. Also, when there appears to be significant overlap between articles fitting concept C, and articles that have been classified by users as belonging to M, and if C does not currently index into M, then M can be added to the list of M indexed into by C. The foregoing heuristic for expanding the concepts C that are covered by M, can potentially make M too broad and, thus, accept too many articles. Therefore, it further is preferred that a reasonable but arbitrary limit is set on the conceptual size covered by M.

With regard to the detailed analysis of each informon A with respect to the community profile for each M, each A must pass through this analysis for each U subscribing to a particular M, i.e., for each member client in a particular community. After A has passed that stage, it is then filtered at a more personal, member client level for each of those users. The profile and filtering process are very similar for both the community level and the member client level, except that at the community level, the empirical data obtained is for all U who subscribed to M, and not merely an individual U. Other information about the individual U can be used to help the filter, such as what U thinks of what a particular author writes in other Zs that the user reads, and articles that can't be used for the group-level M processing.

FIG. **6** illustrates the development of a profile, and its associated predictors. Typically, regarding the structure of a profile **400**, the information input into the structure can be divided into three broad categories: (1) Structured Feature Information (SFI) **405**; (2) Unstructured Feature Information (UFI) **410**; and (3) Collaborative Input (CI) **415**. Features derived from combinations of these three types act as additional peer-level inputs for the next level of the rating prediction function, called (4) Correlated-Feature, Error-Correction Units (CFECU) **420**. From inputs **405**, **410**, **415**, **420**, learning functions **425***a–d* can be applied to get two computed functions **426***a–d*,**428***a–d* of the inputs. These two functions are the Independent Rating Predictors (IRP) **426***a–d*, and the associated Uncertainty Predictors (UP) **428***a–d*. IRPs **426***a–d* can be weighted by dividing them by their respective UPs **428***a–d*, so that the more certain an IRP **426***a–d* is, the higher its weight. Each weighted IRP **429***a–d* is brought together with other IRPs **429***a–d* in a combination function **427***a–d*. This combination function **427***a–d* can be from a simple, weighted, additive function to a far more complex neural network function. The results from this are normalized by the total uncertainty across all UPs, from Certain=zero to Uncertain=infinity, and combined using the Certainty Weighting Function (CWF) **430**. Once the CWF **430** has combined the IRPs **426***a–d*, it is preferred that result **432** be shaped via a monotonically increasing function, to map to the range and distribution of the actual ratings. This function is called the Complete Rating Predictor (CRP) **432**.

SFI **405** can include vectors of authors, sources, and other features of informon A that may he influential in determining the degree to which A falls into the categories in a given M. UFI **410** can include vectors of important words, phrases, and concepts that help to determine the degree to which A falls into a given M. Vectors can exist for different canonical parts of A. For example, individual vectors may be provided for subject/headings, content body, related information in other referenced informons, and the like. It is preferred that a positive and negative vector exists for each canonical part.

CI **415** is received from other Us who already have seen A and have rated it. The input used for CI **415** can include, for example, "interestingness," credibility, funniness, content value, writing quality, violence content, sexual content, profanity level, business importance, scientific merit, surprise/unexpectedness of information content, artistic quality, dramatic appeal, entertainment value, trendiness/importance to future directions, and opinion agreement. Each CFECU **420** is a unit that can detect sets of specific feature combinations which are exceptions in combination. For example, author X's articles are generally disliked in the Z for woodworking, except when X writes about lathes. When an informon authored by X contains the concept of "lathes," then the appropriate CFECU **420** is triggered to signal that this is an exception, and accordingly a signal is sent to offset the general negative signal otherwise triggered because of the general dislike for X's informons in the woodworking Z.

As an example the form of Structured Feature Information (SFI) **405** can include fields such as Author, Source, Information-Type, and other fields previously identified to be of particular value in the analysis. For simplicity, the exemplary SFI, helow, accounts only for the Author field. For this example, assume three authors A, B, and C, have collectivcly submitted 10 articles that have been read, and have been rated as in TABLE 1 (following the text of this specification. In the accompanying rating scheme, a rating can vary between 1 and 5, with 5 indicating a "most interesting" article. If four new articles (**11, 12, 13, 14**) arrive that have not yet been rated, and, in addition to authors A, B, C, and a new author D has contributed, a simple IRP for the Author field, that just takes sums of the averages, would be as follows:

IRP(author)=weighted sum of
- average(ratings given the author so far)
- average(ratings given the author so far in this M)
- average(ratings given all authors so far in this M)
- average(ratings given all authors)
- average(ratings given the author so far by a particular user U)*
- average(ratings given the author so far in this M by a particular user U)*
- average(ratings given all authors so far in this M by a particular user U)*
- average(ratings given all authors by a particular user)*

* (if for a personal Z)

The purpose of the weighted sum is to make use of broader, more general statistics, when strong statistics for a particular user reading an informon by a particular author, within a particular Z may not yet be available. When stronger statistics are available, the hroader terms can be eliminated by using smaller weights. This weighting scheme is similar to that used for creating CWFs **430**, for the profiles as a whole. Some of the averages may be left out in the actual storage of the profile if, for example, an author's average rating for a particular M is not "significantly" different from the average for the author across all Ms. Here, "significance" is used is in a statistical sense, and frameworks such as the Minimum Description Length (MDL) Principle can be used to determine when to store or use a more "local" component of the IRP. As a simple example, the following IRP employs only two of the above terms:

IRP(author)=weighted sum of
- average (ratings given this author so far in this M)
- average (ratings given all authors so far in this M)

Table 2 gives the values attained for the four new articles.

It is preferred that an estimate of the uncertainty resulting from a positive or negative IRP be made, and a complex neural net approach could be used. However, a simpler method, useful for this example, is simply to repeat the same process that was used for the IRP but, instead of predicting the rating, it is preferred to predict the squared-error, given the feature vector. The exact square-error values can be used as the informon weights, instead of using a rating-weight lookup table. A more optimal mapping function could also be computed, if indicated by the application.

	Token 1	Token 2	Token 3	Token 4
IRP pos. vector	16.68	8.73	12.89	11.27
IRP neg. vector	15.20	8.87	4.27	5.04

The UPs then can be computed in a manner similar to the IRP's: comparisons with the actual document vectors can be made to get a similarity measure, and then a mapping function can be used to get an UP.

Making effectivc usc of collaborative input (CI) from other users U is a difficult problem because of the following seven issues. First, there generally is no a priori knowledge regarding which users already will have rated an informon A, before making a prediction for a user U, who hasn't yet read informon A. Therefore, a model for prediction must be operational no matter which subset of the inputs happen to be available, if any, at a given time. Second, computational efficiency must be maintained in light of a potentially very large set of users and informons. Third, incremental updates of rating predictions often are desired, as more feedback is reported from users regarding an informon. Fourth, in learning good models for making rating predictions, only very sparse data typically is available for each users rating of each document. Thus, a large "missing data" problem must be dealt with effectively.

Fifth, most potential solutions to the CI problem require independence assumptions that, when. grossly violated, give very poor results. As an example of an independence assumption violation, assume that ten users of a collaborative filtering system, called the "B-Team," always rate all articles exactly in the same way, for example, because they think very much alike. Further assume that user A's ratings are correlated with the B-Team at the 0.5 level, and are correlated with user C at the 0.9 level. Now, suppose user C reads an article and rates it a "5". Based on that C's rating, it is reasonable to predict that A's rating also might be a "5". Further, suppose that a member of the B-Team reads the article, and rates it a "2". Existing collaborative filtering methods arc likely to predict that A's rating RA would be:

$$R_A=(0.9\times5+0.5\times2)/(0.9+0.5)=3.93$$

In principle, if other members of the B-Team then read and rate the article, it should not affect the prediction of A's rating, R_A, because it is known that other B-Team members always rate the article with the same value as the first

memher of the B-Team. However, the prediction for A hy existing collahorative filtering schemes would tend to give 10 times the weight to the "2" rating, and would be:

$$R_A=(0.9\times5+10\times0.5\times2)/(0.9+10\times0.5)=2.46$$

Existing collahorative filtering schemes do not work well in this case because B-Team's ratings are not independent, and have a correlation among one another of 1. The information filter according to the present invention can recognize and compensate for such inter-user correlation.

Sixth, information about the community of people is known, other than cach user's ratings of informons. This information can include the present topics the users like, what authors the users like, etc. This information can make the system more effective when it is used for learning stronger associations between community memhers. For example, because Users A and B in a particular community M have never yet read and rated an informon in common, no correlation between their likes and dislikes can be made, hased on common ratings alone. However, users A and B have hoth read and liked several informons authored hy the same author, X, although Users A and B each read a distinctly different Zs. Such information can he used to make the inference that there is a possible relationship hetween user A's interests and user B's interests. For the most part, existing collahorative filtering systems can not take advantage of this knowledge.

Seventh, information about the informon under consideration also is known, in addition to the ratings given it so far. For example, from knowing that informon A is ahout the concept of "gardening", hetter use can be made of which users' ratings are more relevant in the context of the information in the informon. If user B's rating agrees with user D's rating of articles when the subject is about "politics", hut B's ratings agree more with user D when informon A is about "gardening", then the relationship between User B's ratings and User D's ratings are preferred to be emphasized to a greater extent than the relationship between User B and User C when making predictions ahout informon A.

With regard to the aforementioned fourth, sixth and seventh issues namely, making effective use of sparse, but known, information about the community and the informon, it is possible to determine the influcnce of user A's rating of an informon on the predicted rating of the informon for a second user, B. For example, where user A and user B have read and rated in common a certain numher of informons, the influence of user A's rating of informon D on the predicted rating of informon D for user B can hc definecl hy a relationship that has two components. First, there can he a common "mindset," S_M hetween user A and user B and informon D, that may he expressed as:

M_S=profile(A) X profile(B) X DocumentProfile(D).

Second, a correlation may he taken hetween user A's past ratings and user B's past ratings with respect to informons that are simdar to D. This correlation can he taken hy weighting all informons E that A and B have rated in common by the similarity of E to D, S_{ED}:

S_{ED}=Weighted_Correlation(ratings(A),ratings(B))

Each of the examples can he weighted hy

$$W_{pr} = \text{weight for rating pair (rating } (A, D), \text{ rating } (B, D))$$

$$= DocumentProfile\,(E) \times DocumentProfile(D)$$

Note that thc "X" in thc ahovc cquation may not be a mere multiplication or cross-product, but rather be a method for comparing the similarity between the profiles. Next, the similarity of the member clicnt profiles and informon content profiles can be compared. A neural network could he used to learn how to compare profiles so that the error in predicted ratings is minimized. However, the invention can be emhodied with use of a simple cosine similarity metric, like that previously considered in connection with Unstructured Feature Information (UFI) can be used.

The method used is preferred to be able to include more than just the tokens, such as the author and other SFI; and, it is preferred that the three vectors for component also are able to be compared. SFIs may he handled by transforming them into an entity that can he treated in a comparahle way to token frequencies that can be multiplied in the standard token frequency comparison method, which would be recognized hy a skilled artisan.

Continuing in thc ongoing example, the Author field may he used. Where user A and user B have rated authors K and L, the token frequency vector may appear as follows:

User	Avg. Rating Given to Author K	# in sample	Avg. Rating Given to Author L	# in sample	Avg. Rating Given to Author M	# in sample
A	3.1	21	1.2	5	N/A	0
B	4	1	1.3	7	5	2

Further, the author component of the member client profiles of user A and user B may be compared by taking a special weighted correlation of each author under comparison. In general, the weight is a function F of the sample sizes for user A's and user B's rating of the author, where F is the product of a monotonically-increasing function of the sample size for each of user A and user B. Also, a simple function G of whether the informon D is hy the author or not is used. This function can be: G=q if so, and G=p<q if not, where p and q are optimized constraints according to the domain of the filtering system. When there has heen no rating of an author hy a user, then the function of thc zero sample size is positive. This is because the fact that the user did not read anything hy the author can signify some indication that the author might not produce an informon which would he highly rated hy the user. In this case, the exact value is an increasing function H of the total articles read by a particular user so far, because it hecomes more likely that the user is intentionally avoiding reading informons hy that author with each subsequent article that has been read but is not prepared by the author. In general, the exact weighting function and parameters can be empirically derived rather than theoretically derived, and so is chosen by thc optimization of thc overall rating prediction functions. Continuing in the present example, a correlation can be computed with the following weights for the authors K, L and M.

Author	Weight
K	F(21,1,not author) = log(21 + 1) × log(1 + 1) × G(not author) = 0.04
L	F(5,7, author or D) = log(5 + 1) × log(7 + 1) × G(author) = 0.70

-continued

Author	Weight
M	F(0.2, not author) = H(26) × log(2 + 1) × G(not author) = 0.02

It is preferred that the logarithm be used as the monotonically-increasing function and that p=1, q=0.1. Also used are H=log(sample$_{size}$.0.1) and an assumed rating, for those authors who are unrated by a user, to the value of "2." The correlation for the author SFI can be mapped to a non-zero range, so that it can be included in the cosine similarity metric. This mapping can be provided by a simple one-neuron neural network, or a linear function such as, (correlation+1)*P_0. Where the P_0 is an optimized parameter used to produce the predicted ratings with the lowest error in the given domain for filtering.

An artisan skilled in information retrieval would recognize that there are numerous methods that can be used to effect informon comparisons, particularly document comparisons. One preferred method is to use a TF-IDF weighting technique in conjunction with the cosine similarity metric. SFI including author, can be handled by including them as another token in the vector. However, the token is preferred to be weighted by a factor that is empirically optimized rather than using a TF-IDF approach. Each component of the relationship between user A's and user B's can be combined to produce the function to predict the rating of informon D for user B. The combination function can be a simple additive function, a product function, or a complex function, including, for example, a neural network mapping function, depending upon computational efficiency constraints encountered in the application. Optimization of the combination function can be achieved by minimizing the predicted rating error as an objective.

In addition to determining the relationship between two user's ratings, a relationship that can be used and combined across a large population of users can be developed. This relationship is most susceptible to the aforementioned first, second, third, and fifth issues in the effective use of collaborative input. Specifically, the difficulty with specifying a user rating relationship across a large population of users is compounded by the lack of a priori knowledge regarding a large volume of dynamically changing information that may have unexpected correlation and therefore grossly violate independence assumptions.

In one embodiment of the present invention, it is preferred that users be broken into distributed groups called "mindpools." Mindpools can be purely hierarchical, purely parallel, or a combination of both. Mindpools can be similar to the aforementioned "community" or may instead be one of many subcommunities. These multiple hierarchies can be used to represent different qualities of an article. Some qualities that can be maintained in separate hierarchies include: interestingness; credibility; funniness; valuableness; writing quality; violence content; sexual content; profanity level; business importance; scientific merit; artistic quality; dramatic appeal; entertainment value; surprise or unexpectedness of information content; trendiness or importance to future directions; and opinion agreement. Each of these qualities can be optionally addressed by users with a rating feedback mechanism and, therefore, these qualities can be used to drive separate mind pool hierarchies. Also, the qualities can be used in combinations, if appropriate, to develop more complex composite informon qualities, and more sublime mindpools.

FIG. **7** illustrates a preferred embodiment of a mind pool system **500**. It is preferred that all users be members of the uppermost portion of the hierarchy, namely, the top mind pool **501**. Mind pool **501** can be broken into sub-mindpools **502***a–c*, which separate users into those having at least some common interests. Furthermore, each sub-mind pool **502***a–c* can be respectively broken into sub-sub-mindpools **503***a–b*, **503***c–d*, **503***e,f,g* to which users **504***a–g* are respective members. As used herein, mind pool **501** is the parent node to sub-mindpools **502***a–c*,and sub-mindpools **502***a–c* are the respective parent nodes to sub-sub-mindpools **503***a–g*. Sub-pools **502***a–c* are the child nodes to mind pool **501** and sub-pools **503***a–g* are child nodes to respective mindpools **503***a–c*. Sub-pools **503***a–g* can be considered to be end nodes. Users **505***a,b* can be members of sub-mind pool **502***a*, **502**c, if such more closely matches their interests than would membership in a sub-sub-mind pool **503***a–g*. In general, the objective is to break down the entire population of users into subsets that are optimally similar. For example, the set of users who find the same articles about "gardening" by author A to be interesting but nevertheless found other articles by author A on "gardening" to be uninteresting may be joined in one subset.

A processing means or mind pool manager may be used to handle the management of each of the mindpools **501**, **502***a–c*, and **503***a–g*. A mind pool manager performs the following functions: (1) receiving rating information from child-node mind pool managers and from those users coupled directly to the manager; (2) passing rating information or compiled statistics of the rating information up to the manager's parent node, if such exists; (3) receiving estimations of the mind pool consensus on the rating for an informon from the manager's parent mind pool, if such exists; and (4) making estimations of the mind pool consensus on the rating for a specific informon for the users that come under the manager's domain; and (5) passing the estimations from function **4** down to either a child-node mind pool or, if the manager is an end node in the hierarchy, to the respective user's CWF, for producing the user's predicted rating. Function **4** also can include combining the estimations received from the manager's parent node, and Uncertainty Predictions can be estimated based on sample size, standard deviation, etc. Furthermore, as alluded to above, users can be allowed to belong to more than one mind pool if they don't fit precisely into one mind pool but have multiple views regarding the conceptual domain of the informon. Also, it is preferred that lateral communication be procided between peer managers who have similar users beneath them to share estimation information. When a rating comes in from a user, it can be passed to the immediate manager(s) node above that user. It is preferred that the manager(s) first decide whether the rating will effect its current estimation or whether the statistics should be passed upward to a parent-node. If the manager estimation would change by an amount above an empirically-derived minimum threshold, then the manager should pass that estimation down to all of its child-nodes. In the event that the compiled statistics are changed by more than another minimum threshold amount, then the compiled statistics should be passed to the manager's parent-node, if any ,and the process recurses upward and downward in the hierarchy.

Because no mind pool manager is required to have accurate information, but just an estimation of the rating and an uncertainty level, any manager may respond with a simple average of all previous documents, and with a higher degree of uncertainty, if none of its child-nodes has any rating information yet. The preferred distributed strategy

tends to reduce the communication needed between processors, and the computation tends to be pooled, thereby eliminating a substantial degree of redundancy. Using this distributed strategy, the estimations tend to settle to the extent that the updating of other nodes, and the other users predictions are minimized. Therefore, as the number of informons and users becomes large, the computation and prediction updates grow as the sum of the number of informons and the number of users, rather than the product of the number of informons and the number of users. In addition, incremental updates can be accomplished by the passing of estimations up and down the hierarchy. Incremental updates of rating predictions continue to move until the prediction becomes stable due to the large sample size. The distributed division of users can reduce the effects of independent assumption violations. In the previous example with the B-Team of ten users, the B-Team can be organized as a particular mind pool. With the additional ratings from each of the B-Team members, the estimation from the B-Team mind pool typically does not change significantly because of the exact correlation between the members of that mind pool. This single estimation then can be combined with other estimations to achieve the desired result, regardless of how many B-Team members have read the article at any given time.

The mind pool hierarchies can be created by either computer- or human-guided methods. If the hierarchy creation is human-guided, there often is a natural breakdown of people based on information such as job position, common interests, or any other information that is known about them. Where the mind pool hierarchy is created automatically, because the previously described measure of the collaborative input relationship between users can be employed in a standard hierarchical clustering algorithm to produce each group of users or nodes in the mind pool hierarchy. Such standard hierarchical clustering algorithms can include, for example, the agglomerative method, or the divide-and-conquer method. A skilled artisan would recognize that many other techniques also are available for incrementally-adjusting the clusters as new information is collected. Typically, clustering is intended to (1) bring together users whose rating information is clearly not independent; and (2) produce mind pool estimations that are substantially independent among one another.

Estimations are made in a manner similar to other estimations described herein. For example, for each user or sub-mind pool (sub-informant), a similarity between the sub-informant and the centroid of the mind pool can be computed in order to determine how relevant the sub-informant is in computing the estimation. Uncertainty estimators also are associated with these sub-informants, so that they can be weighted with respect to their reliability in providing the most accurate estimation. Optionally, the informon under evaluation can be used to modulate the relevancy of a sub-informant. This type of evaluation also can take advantage of the two previously-determined collaborative information relationship components, thereby tending to magnify relationships that are stronger for particular types of informons than for others. Once a suitable set of weights are established for each user within a mind pool for a particular informon, a simple weighted-average can be used to make the estimation. It is preferred that the "simple" weighted average used is more conservative regarding input information that a simple independent linear regression. Also, the overall Uncertainty can be derived from the Uncertainty Predictions of the sub-informants, in a manner similar to the production of other uncertainty combination methods described above. Approximations can be made by pre-computing all terms that do not change significantly, based on the particular informon, or the subset of actual ratings given so far to the mind pool manager.

As stated previously, the correlated-feature error-correction units (CFECUs) are intended to detect irregularities or statistical exceptions. Indeed, two objectives of the CFECU units are to (1) find non-linear exceptions to the general structure of the three aforementioned types of inputs (SFI, UFI, and CI); and (2) find particular combinations of informon sub-features that statistically stand out as having special structure which is not captured by the rest of the general model; and (3) trigger an additional signal to the CFECU's conditions are met, in order to reduce prediction error. The following exemplifies the CFECU operation.

	User B's Avg. Rating of of Informons About	
	Gardening	Politics
Author A's Articles	4.5	1.2
Other Authors	1.4	2
Weighted by Topic	1.68	1.87

	User B's number of Informons Read About		Average over
	Gardening	Politics	Topics
Author A's Articles	7	40	1.69
Other Authors	70	200	1.84

In this example, it is desired that author A's informon D about gardening have a high predicted rating for user B. However, because the average rating for author A by user B is only 1.69, and the average rating for the gardening concept is only 1.68, a three-part model (SFI-UFI-CI) that does not evaluate the informon features in combination would tend to not rank informon D very highly. In this case, the first CFECU would first find sources of error in past examples. This could include using the three-part model against the known examples that user B has rated so far. In this example, seven articles that user B has rated, have an average rating of 4.5, though even the three-part model only predicts a rating of about 1.68. When such a large error appears, and has statistical strength due to the number of examples with the common characteristics of, for example, the same author and topic, a CFECU is created to identify that this exception to the three-part model has been triggered and that a correction signal is needed. Second, it is preferred to index the new CFECU into a database so that, when triggering features appear in an informon, for example, author and topic, the correction signal is sent into the appropriate CWF. One method which can be used to effect the first step is a cascade correlation neural network, in which the neural net finds new connection neural net units to progressively reduce the prediction error. Another method is to search through each informon that has been rated but whose predicted rating has a high error, and storing the informons profile.

When "enough" informons have been found with high error and common characteristics, the common characteristics can be joined together as a candidate for a new CFECU. Next, the candidate can be tested on all the samples, whether

they have a high prediction or a low prediction error associated with them. Then, the overall error change (reduction or increase) for all of the examples can be computed to determine if the CFECU should be added to the informon profile. If the estimated error reduction is greater than a minimum threshold level, the CFECU can be added to the profile. As successful CFECU are discovered for users'profiles, they also can be added to a database of CFECU's that may be useful for analyzing other profiles. If a particular CFECU has a sufficiently broad application, it can be moved up in the filtering process, so that it is computed for every entity once. Also, the particular CFECU can be included in the representation that is computed in the pre-processing stage as a new feature. In general, the estimation of the predicted rating from a particular CFECU can be made by taking the average of those informons for which the CFECU responds. Also, the Uncertainty can be chosen such that the CFECU signal optimally outweigbs tbe otber signals being sent to the CWF. One method of self-optimization that can be employed is, for example, the gradient descent method, although a skilled artisan would recognize that other appropriate optimization methods may be used.

The invention of this continuation-in-part application, as shown in FIGS. **8** and **9**, provides a collaborative and preferably adaptive search engine system in which elements of the structure and principles of operation of the apparatus of FIGS. **1–7** are applied. Accordingly, a searcb engine system of the invention, as preferably embodied, integrates collaborative filtering with adaptive content-based filtering to provide improved searcb engine performance. The acronym "CASE" refers to a search engine system of the invention, i.e., a collaborative, adaptive search engine.

In the operation of conventional search engines at portal web sites., user queries are searched on demand to find relevant informons across the web. Content-based filtering is typically used in measuring the relevacy of informons, and the search results are resented in the form of a list of informons ranked by relevancy.

The present invention combines collaborative filtering with content-based filtering in measuring informons for relevancy, and further preferably applies adaptive updating of the content-based filtering operation. In providing these results, tbe invention can be embodied as a search engine system in accordance with different basic structures. In the presently preferred basic structure, an integrated collaborative/content-based filter (FIGS. **1–7**) is operated to provide ongoing or continuous searching for selected user queries, with a "wire" being established for eacb query. On the other hand, a regular search engine is operated to make immediate or short-term "demand" searches for other user queries on tbe basis of content-based filtering. This basic structure of the invention is especially beneficial for use in applying the invention to existing search engine structure.

Demand searcb results can be returned if no wire exists for an input query. Otherwise, wire search results are returned if a wire does exist, or collaborative ranking data can he applied from the wire filter structure to improve the results of the demand search from the regular search engine.

In the currently preferred embodiment, wires are created for the most common queries received by the search engine system. A suitable analysis is applied to the search engine operations to determine which queries are most common, and respective wires are then created for each of these queries. An analysis update can be made from time to time to make wire additions or deletions as warranted.

When a user makes a query for which a wire already exists, wire search results are preferably returned instead of regular search engine results. As shown in the logic diagram of FIG. **7**, a user provides a query as indicated by block **20C**. The query is applied to a Lookup Table, as indicated by block **22C**, block **24C** applies a test to determine from the table whether a wire already exists for the new query. If so, block **26C** returns results from tbe existing wire. Otberwise, block **28C** commands a demand search by a regular query engine.

With the use of wire search returns, each user can review tbe returned results and provide feedback data about reviewed documents. Sucb feedback data is incorporated in the filter profiles used in processing informons for the wire. Therefore, when a future user makes substantially the same query, the wire will bave been improved by the incorporation of previous users' feedback data. By analyzing documents wbich users rate as meeting a particular quality sucb as interestingness, the system can find common document features which can be used to return more like documents to future users who make substantially the same query.

Alternatively, all queries applied to a search engine system of the invention can set up new wires. After a search query is presented to the search engine system, a wire is created on the basis of the query terms, and all new documents subsequently received from the network are filtered by the new wire. A push-model may be used to send all passed, new documents to the user.

Among otber basic searcb engine system structures, an integrated system can be employed in which collaborative and content-based filtering is structured to provide demand searches with or witbout collaborative filtering, or wire searches. In the operation of the preferred basic structure and other basic structures, a query processor can be employed, if needed, to make search-type assignments for user queries. Generally, basic search engine system structures of the invention are preferably embodied witb tbe use of a programmed computer system.

Collaborative filtering employs additional data from other users to improve searcb results for an individual user for whom a search is being conducted. The collaborative data can be feedback informon rating data, and/or it can be content-profile data for agent mind melding which is more fully disclosed in Ser. No. 09/195,708 now pending, entitled INTEGRATED COLLABORATIVE/CONTENT-BASED FILTER STRUCTURE EMPLOYING SELECTIVELY SHARED, CONTENT-BASED PROFILE DATA TO EVALUATE INFORMATION ENTITIES IN A MASSIVE INFORMATION NETWORK, filed by the current inventors on Nov. 19, 1998, and bereby incorporated by reference.

Many types of user rating information can be used. For example, users can sort documents which they have read from best to worst. Alternatively, users can select on a scale (numeric, such as 1 to 10, or worded, such as good, medium, poor) how much they enjoyed reading a document. Further, user monitoring can measure time spent by users on each document, thereby indicating user interest (normalized by document length). Among other possibilities, the choices of documents for reading by other users can be simply used as an indication of interesting documents. In all cases, the feedback rating data can be based on interestingness or any of a variety of other document qualities, as described in connection with FIGS. **1–7**.

Feedback ranking information can be used in a number of ways, and the invention is not limited by the method of feedback information use. Use metbods range in spectrum from weighting relative ranks by a set amount (possibly equally, possibly heavy weighting one above the other) to dynamically adjusting the weight by measuring bow statis-

tically significant the user feedhack is. For example, if only one person has ranked an article, it may not be significant. However, if many people have consistently ranked an article the same, more credihility may be placed on the user's weighting.

FIG. **9** shows a generalized embodiment of the invention in which system elements in a CASE system **30C** are integrally configured to provide wire and/or demand searches. A query processor **32C** receives queries from an individual user **34C** and other users **36C**. A mode selector **38C** responds to the currently processed query to set a content-hased filter structure **40C** for wire search operation or demand search operation. In the preferred application of the invention, the wire mode is selected only if a wire already exists, and wires exist only for those queries found to be commonly entered as previously described. In the demand search mode, the filter structure **40C** can function similarly to a normal search engine.

Otherwise, various schemes can be used for determining whether a wire search or a demand search is made. For example, every query can call for a wire search, with a demand search being made the first time a particular query is entered and with wire searches heing made for suhsequent entries of the same query. As another example, the user may select a demand search, or, if continuing network searching is desired, the user may select a wire search.

The filter structure **40C** operates in its set wire search mode or demand search mode, and employs content-hased profiles **42C** in content-hased filtering (preferahly multi-level as described in connection with FIGS. **1–7**). Wire profiles **42C 1** are adaptively updated with infornon-evaluation, feedback data from users respectively associated therewith. These profiles are used by the filter structure **40C** in wire searches in the wire mode.

Demand profiles **42C2** are used by the filter structure **40C** in demand searches in the demand mode. Collaborative profile data can he integrated with the wire profiles through agent mind melding **43C** as previously explained.

A spider system **46C** scans a network **44C** to find informons for a current demand search, and to find informons with continued network scanning for existing wires. In sclecting availahle informons for return, the spider system **46C** uses a content threshold derived from the content-hased profile for which an informon search is being conducted.

In many instances, it s preferahle that the spider system **46C** have a memory system **46CM** which holds an informon data hasc wherein index information is stored from informons previously collected from the network. In this manner, demand searches can be quickly made from the spider memory **46CM** as opposed to making a time consuming search and downloading in response to a search demand query from the search engine.

A search return processor **48C** receives either demand search informons or wire search informons passed by the content-hased filter structure **40C** according to the operating mode of the latter, and includes an informon rating system which is like that of FIG. **6**. The informon rating system combines content-hased filtering data with collaborative feedback rating data, from users through a feedback processor **50C** at least in the wire search mode and, if desired, in the demand search mode.

In the wire search mode, the processor **48C** rates informons on a continuing hasis as they are received from the network **44C** through the spider system **46C** as indicated hy the reference character **48C1**. In the demand search mode, the processor **48C** rates informons returned hy the spider system **46C** in a demand search as indicated hy the reference character **48C2**. Collaborative rating data is used in the informon rating process in the wire search mode, and if applied in the demand search mode, to the extent that collaborative data is availahle for the informons in the search return. Search results are returned to the users **34C** and **36C** from the search return processor **48C** as shown in FIG. **9**.

The invention is preferably embodied as shown in FIG. **10**. A query processor **60C** receives queries from an individual user **62C** and other users **64C** and determines whether a wire already exists for each entered query. If a wire exists, the query is routed to a collahorative/content-hased filter structure **66C** like that of FIGS. **1–7**. A spider system **68C** continuously scans a network **70C** for informons providing a threshold-level match for content hased profiles (i.e., preprocessing profiles at the top level of the preferred multi-level filter structure, at least one of which reflects the content profile of a current wire query). Informons which are passed hy the filter **66C** for existing wires are stored in a memory **72C** according to the wire or wires to which they helong.

A feedhack processor **74C** is structured like the mind pool system of FIG. **7** to provide collaborative feedhack data for integration with the content-hased data in the measurement of informon relevancy hy the filter **66C**. An informon rating structure like that of FIG. **6** is employed for this purpose. Adaptive feedhack data is applied from the users to the filter **66C** as shown in ordcr to update content profiles as previously described.

If no wire exists for a currently input query, the query is sent to a regular search engine where a content profile is established for content hased filtering of informons returned by a spider system **78C** in a demand search of the network **70C**. The spider system **78C** can have its own memory system **78CM** as considered in connection with the spider **46C** of FIG. **9**.

Once filtering is performed on returned informons, those informons which provide a satisfactory match to the query are returned as a list to the user through a search return processor **80C**. The processor **80C** creates a new wire for the current query for which a demand search was made, if a demand search memory **82C** indicates that the current query has heen made over time with sufficient frequency to qualify as a "common" query for which a wire is justified. As indicated by dashed connector line **80FD**, collaborative feedhack data can he, and preferahly is, integrated into the demand search processing hy the processor **80C**.

Many alterations and modifications may he made by those having ordinary skill in the art without departing from the spirit and scope of the invention. Therefore, it must he understood that the illustrated embodiments have been set forth only for the purposes of example, and that it should not be taken as limiting the invention as defined by the following claims. The following claims are, therefore, to be read to include not only the combination of elements which are literally set forth but all equivalent elements for performing substantially the same function in suhstantially the same way to ohtain suhstantially the same result. The claims are thus to be understood to include what is specifically illustrated and described above, what is conceptually equivalent, and also what incorporates the essential idea of the invention.

TABLE 1

Article	Author	Rating given
1	A	5
2	B	1
3	B	2
4	B	5
5	C	2
6	C	2
7	C	1
8	C	2
9	C	2
10	C	2

TABLE 2

Article	Author			normalized			normalized	
IRP (author)		avg (author)		weight	weight	avg (all auth)	weight	weight
11	A	5.00	3.12	0.86	2.40	0.49	0.14	4.65
12	B	2.67	0.23	0.32	2.40	0.49	0.66	2.49
13	C	1.83	6.00	0.92	2.40	0.49	0.06	1.86
14	D	N/A	0.00	0.00	2.40	0.49	1.00	2.40

What is claimed is:

1. A search engine system comprising:

a first system for receiving informons from a network on a continuing search basis, for filtering such informons for relevancy to a query from an individual user, and for storing a ranked list of relevant informons as a wire;

a second system for receiving informons from a network on a current demand search basis and for filtering such informons for relevancy to the query from the individual user; and

a third system for selecting at least one of the first and second systems to make a search for the query and to return the wire or demand search results to the individual user.

2. The system of claim **1** wherein the third system selects the first system to make a wire search only if a wire already exists for the query.

3. The system of claim **1** wherein:

a feedback system is provided for receiving collaborative feedback data from system users relative to informons considered by such users; and

at least the first system combines pertaining data from the feedback system with content profile data of the first system in filtering each informon for relevance to the query and inclusion in the wire.

4. The system of claim **3** wherein the first system includes a multi-level, content-based filter having descending levels including at least an upper preprocessing level, a middle user community level, and a bottom user level.

5. The method of claim **3** wherein the collaborative feedback data comprises active feedback data.

6. The method of claim **3** wherein the collaborative feedback data comprises passive feedback data.

7. The method of claim **6** wherein the passive feedback data is obtained by passively monitoring the actual response to a proposed informon.

8. The method of claim **3** wherein the collaborative feedback data comprises a combination of active feedback data and passive feedback data.

9. The system of claim **1** wherein adaptive user feedback data is applied at least to the first system to provide updating of content profile data employed therein.

10. A search engine system comprising:

a system for scanning a network to make a demand search for informons relevant to a query from an individual user;

a content-based filter system for receiving the informons from the scanning system and for filtering the informons on the basis of applicable content profile data for relevance to the query; and

a feedback system for receiving collaborative feedback data from system users relative to informons considered by such users;

the filter system combining pertaining feedback data from the feedback system with the content profile data in filtering each informon for relevance to the query.

11. The system of claim **10** wherein adaptive user feedback data is applied to the content-based filter to provide a learning component for content profile data employed therein.

12. The system of claim **10** wherein:

the scanning system further scans the network on a continuing basis to make a wire search for informons relevant to wire queries from system users; and

the filter system combines pertaining feedback data from the feedback system with applicable content profile data in filtering each wire informon for relevance to applicable wire query.

13. The system of claim **10** wherein the collaborative feedback data comprises active feedback data.

14. The system of claim **10** wherein the collaborative feedback data comprises passive feedback data.

15. The system of claim **14** wherein the passive feedback data is obtained by passively monitoring the actual response to a proposed informon.

16. The system of claim **10** wherein the collaborative feedback data comprises a combination of active feedback data and passive feedback data.

17. A search engine system comprising:

a content-based filtering system for receiving informons from a network on a continuing basis and for filtering the informons for relevancy to a wire or demand query from an individual user;

a feedback system providing feedback data from other users;

a system for controlling the operation of the filtering system to filter for one of a wire response and a demand response and to return the one response to the user; and

the filtering system combining pertaining feedback data from the feedback system with content profile data in determining the relevancy of the informons for inclusion in at least a wire response to the query.

18. The system of claim **17** wherein:

the content-based filtering system includes a collaborative/content based filter for filtering informons for relevancy to a wire query on a continuing basis; and

the content-based filtering system includes a regular search engine for filtering informons for relevancy to a demand query.

19. The system of claim **18** wherein adaptive user feedback data is applied at least to the collaborative/content-based filter to provide learning for content profile data employed therein.

20. The search engine system of claim **17** wherein the feedback system provides active feedback data.

21. The search engine system of claim **17** wherein the feedback system provides passive feedback data.

22. The search engine system of claim **21** wherein the passive feedback data is obtained by passively monitoring the actual response to a proposed informon.

23. The system of claim **17** wherein the feedback system provides a combination of active feedback data and passive feedback data.

24. A method for operating a search engine system comprising:

receiving informons in a first system from a network on a continuing search basis, for filtering such informons for relevancy to a query from an individual user and for storing a ranked list of relevant informons as a wire;

receiving informons in a second system from a network on a current demand search basis for filtering such informons for relevancy to the query from the individual user; and

selecting at least one of the first and second systems to make a search for the query and to return the wire or demand search results to the individual user.

25. A method for operating a search engine system comprising:

scanning a network to make a demand search for informons relevant to a query from an individual user;

receiving the informons in a content-based filter system from the scanning system and filtering the informons on the basis of applicable content profile data for relevance to the query;

receiving collaborative feedback data from system users relative to informons considered by such users; and

combining pertaining feedback data with the content profile data in filtering each informon for relevance to the query.

26. The method of claim **25** wherein the collaborative feedback data comprises active feedback data.

27. The method of claim **25** wherein the collaborative feedback data provides passive feedback data.

28. The method of claim **27** wherein the passive feedback data is obtained by passively monitoring the actual response to a proposed informon.

29. The method of claim **25** wherein the collaborative feedback data comprises a combination of active feedback data and passive feedback data.

30. A method for operating a search engine system comprising:

receiving informons in a content-based filtering system from a network on a continuing basis and filtering the informons for relevancy to a wire or demand query from an individual user;

providing feedback data from other users;

controlling the operation of the filtering system to filter for one of a wire response and a demand response and to return the one response to the user; and

combining pertaining feedback data with content profile data in the filtering system in determining the relevancy of the informons for inclusion in at least a wire response to the query.

31. The method of claim **30** wherein the step of providing feedback data comprises providing active feedback data.

32. The method of claim **30** wherein the step of providing feedback data comprises providing passive feedback data.

33. The method of claim **32** wherein the passive feedback data is obtained by passively monitoring the actual response from at least one of the other users to a proposed informon.

34. The method of claim **30** wherein the step of providing feedback data comprises providing a combination of active feedback data and passive feedback data.

35. A search engine system comprising:

means for receiving informons from a network on a continuing search basis, for filtering such informons for relevancy to a query from an individual user, and for storing a ranked list of relevant informons as a wire;

means for receiving informons from a network on a current demand search basis and for filtering such informons for relevancy to the query from the individual user; and

means for selecting at least one of the first and second systems to make a search for the query and to return the wire or demand search results to the individual user.

36. A search engine system comprising:

means for content-based filtering informons received from a network on a continuing basis for relevancy to a wire or demand query from an individual user;

means for collecting feedback data from other users;

means for controlling the operation of the filtering means to filter for one of a wire response and a demand response and to return the one response to the user; and

the filtering means combining pertaining feedback data from the feedback system with content profile data in determining the relevancy of the informons for inclusion in at least a wire response to the query.

* * * * *

US006094649A

United States Patent [19]

Bowen et al.

[11] **Patent Number: 6,094,649**

[45] **Date of Patent: Jul. 25, 2000**

[54] **KEYWORD SEARCHES OF STRUCTURED DATABASES**

[75] Inventors: **Stephen J Bowen**, Sandy; **Don R Brown**, Salt Lake City, both of Utah

[73] Assignee: **PartNet, Inc.**, Salt Lake City, Utah

[21] Appl. No.: **08/995,700**

[22] Filed: **Dec. 22, 1997**

[51] **Int. Cl.**[7] **G06F 17/30**

[52] **U.S. Cl.** **707/3**; 707/5; 707/4

[58] **Field of Search** 707/1, 2, 3, 4, 707/5, 531, 532, 500

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,375,235	12/1994	Berry et al.	707/5
5,469,354	11/1995	Hatakeyama et al.	707/3
5,546,578	8/1996	Takada	707/5
5,685,003	11/1997	Peltonen et al.	707/531
5,787,295	7/1998	Nakao	707/500
5,787,421	7/1998	Nomiyama	707/5
5,799,184	8/1998	Fulton et al.	707/2
5,832,479	11/1998	Berkowitz et al.	707/3
5,845,273	12/1998	Jindal	707/1
5,845,305	12/1998	Kujiraoka	707/532
5,848,409	12/1998	Ahn	707/3
5,848,410	12/1998	Walls et al.	707/4

OTHER PUBLICATIONS

"AltaVista Software—Press Release", Anon., *AltaVista Software*, 1997, p. 1.

"Assorted References", pp. 1–38.

"Charles Schwab Broadens Deployment of Fulcrum–Based 'Corporate Knowledge' Library Application", Uknown, Fulcrum Technologies Inc., Mar. 3, 1997, pp. 1–3.

(List continued on next page.)

Primary Examiner—Hosain T. Alam

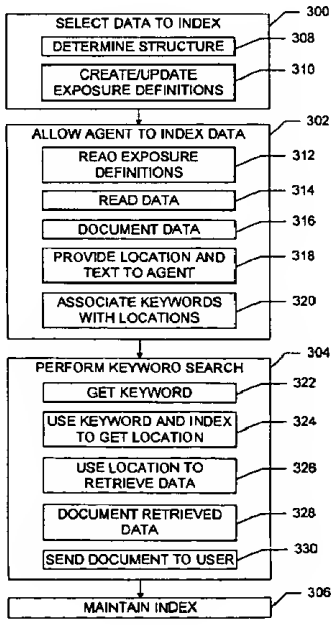
Assistant Examiner—Thuy Pardo

Attorney, Agent, or Firm—Computer Law++

[57] **ABSTRACT**

Methods and systems are provided for supporting keyword searches of data items in a structured database, such as a relational database. Selected data items are retrieved using an SQL query or other mechanism. The retrieved data values are documented using a markup language such as HTML. The documents are indexed using a web crawler or other indexing agent. Data items may be selected for indexing by identifying them in a data dictionary. The indexing agent produces an index that associates keywords with resource locators such as URLs, hot links, file paths, or distinguished names. After a user provides a keyword to a search engine interface, the index is used to obtain a resource locator that is associated with the keyword. The resource locator is used to retrieve the item's current data from the structured database. A document containing the retrieved data is then generated and provided to the user.

45 Claims, 3 Drawing Sheets



OTHER PUBLICATIONS

"Deployment Choices for English Wizard", Anon., Linguistic Technology Corporation, 1996, pp. 1–2.
"Effective Use of Relational Databases On The Internet", L. Harris, Linguistic Technology Corporation, 1996, pp. 1–3.
"Exposé", Uknown, Linguistic Technology Corporation, 1996, pp. 1–2.
"Fulcrum Corporate Overview", Unknown, Fulcrum Technologies Inc., 1997, pp. 1–6.
"Fulcrum Knowledge Network", Unknown, Fulcrum Technologies Inc., 1997, pp. 1–10.
"Fulcrum SearchServer", Unknown, Fulcrum Technologies Inc., 1995–1996, pp. 1–4.
"Fulcrum unifies data searches", J. Senna, InfoWorld Publishing Company, Apr. 28, 1997, pp. 1–2.
"Independent Market Research Ranks Fulcrum 'Number One'", Uknown, Fulcrum Technologies Inc., Jun. 17, 1997, pp. 1–2.
"INFORMIX–Universal Web Connect: Getting Started", Uknown, www.informix.com, no later than Nov. 14, 1997, pp. 1–10.
"Introduction to ALIWEB", Uknown, NEXOR Ltd, 1995, p. 1.
"Knowledge Network: Fulcrum's Leading Edge Technology", J. Blair, Gartner Group, Mar. 26, 1997, pp. 1–2.
"Managing Text with Oracle8 ConText Cartridge", Uknown, Oracle Corporation, 1997, pp. 1–10.
"Nabisco Selects Fulcrum Find! For Information Sharing Across The Organization", Uknown, Fulcrum Technologies Inc., Feb. 3, 1997, pp. 1–3.
"Oracle ConText® Cartridge Release 2.0" Unknown, Oracle Corporation, 1995, 1997, pp. 1–4.
"Plain–English Database tools—English Wizard and VB ELF let you make database queries without using SQL", A. Feibus, CMP Media Inc., Nov. 17, 1997, pp. 1–5.
"SEARCH '97 White Paper", P. Courtot, www.verity,com, no later than Jun. 6, 1997, pp. 1–6.
"Site–index.pl—indexing your Web site", R. Thau, www.ai.mit.edu, no later than Nov. 13, 1997, pp. 1–4.
"Strategic Direction in Electronic Commerce and Digital Libraries: Towards a Digital Agora", N. Adam et al., *ACM Computing Surveys, vol. 28, No. 4*, Dec. 1996, pp. 818–835.
"Sybase SQL Anywhere Professional and the Internet", Uknown, Sybase, Inc., 1997, pp. 1–8.
"Text–enabling Web Applications with Oracle ConText Option", Uknown, Oracle Corporation, 1995, 1997, pp. 1–8.
"The TSIMMIS Approach to Mediation: Data Models and Languages", H. Garcia–Molina et al., Stanford University, Unknown, pp. 1–17.
"Unlocking the Value of Text with Oracle ConText Cartridge", F. Litman, Oracle Corporation, 1994–97, pp. 1–3.
"The Web Robots Database", M. Koster, info.webcrawler-.com, no later than Nov. 13, 1997, pp. 1–2.
"What tools are currently availahle", L. Cooper, stork.uk-c.ac.uk, no later than Nov. 13, 1997, pp. 1–2.
"Yahoo!", uknown, Yahoo! Inc., 1994–97, p. 1.

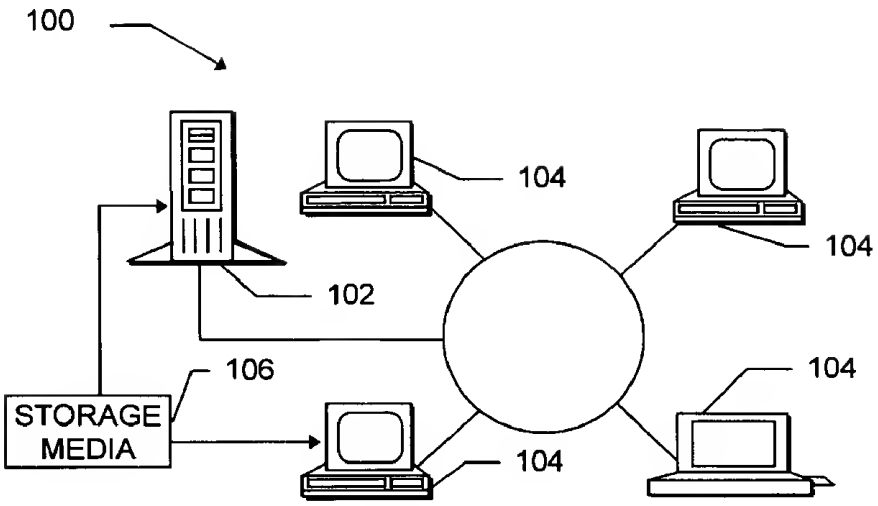

100
104
104
102
106
104
STORAGE MEDIA
104

FIG. 1

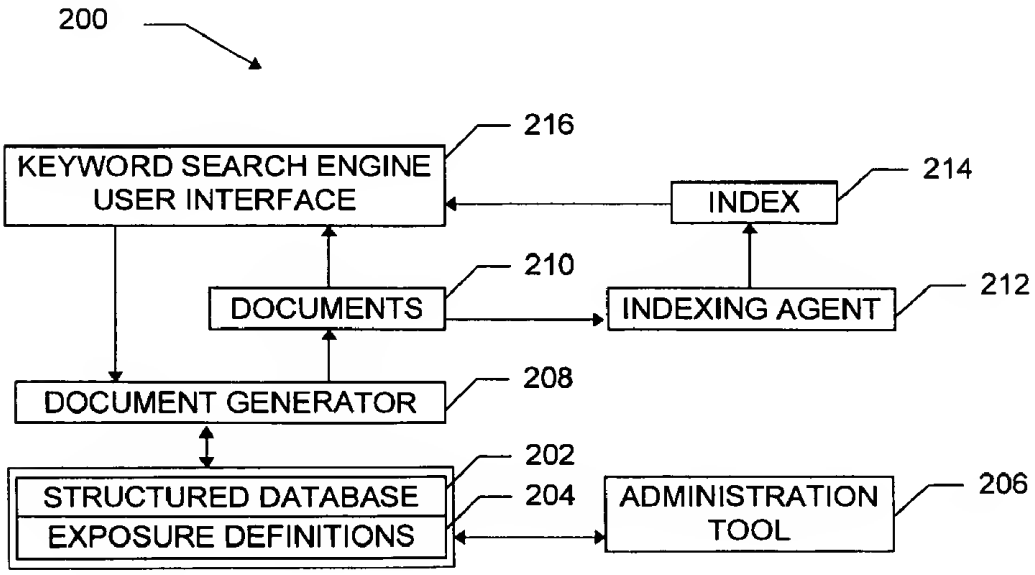

200
216
KEYWORD SEARCH ENGINE USER INTERFACE
214
INDEX
210
DOCUMENTS
212
INDEXING AGENT
208
DOCUMENT GENERATOR
202
STRUCTURED DATABASE
204
EXPOSURE DEFINITIONS
ADMINISTRATION TOOL
206

FIG. 2

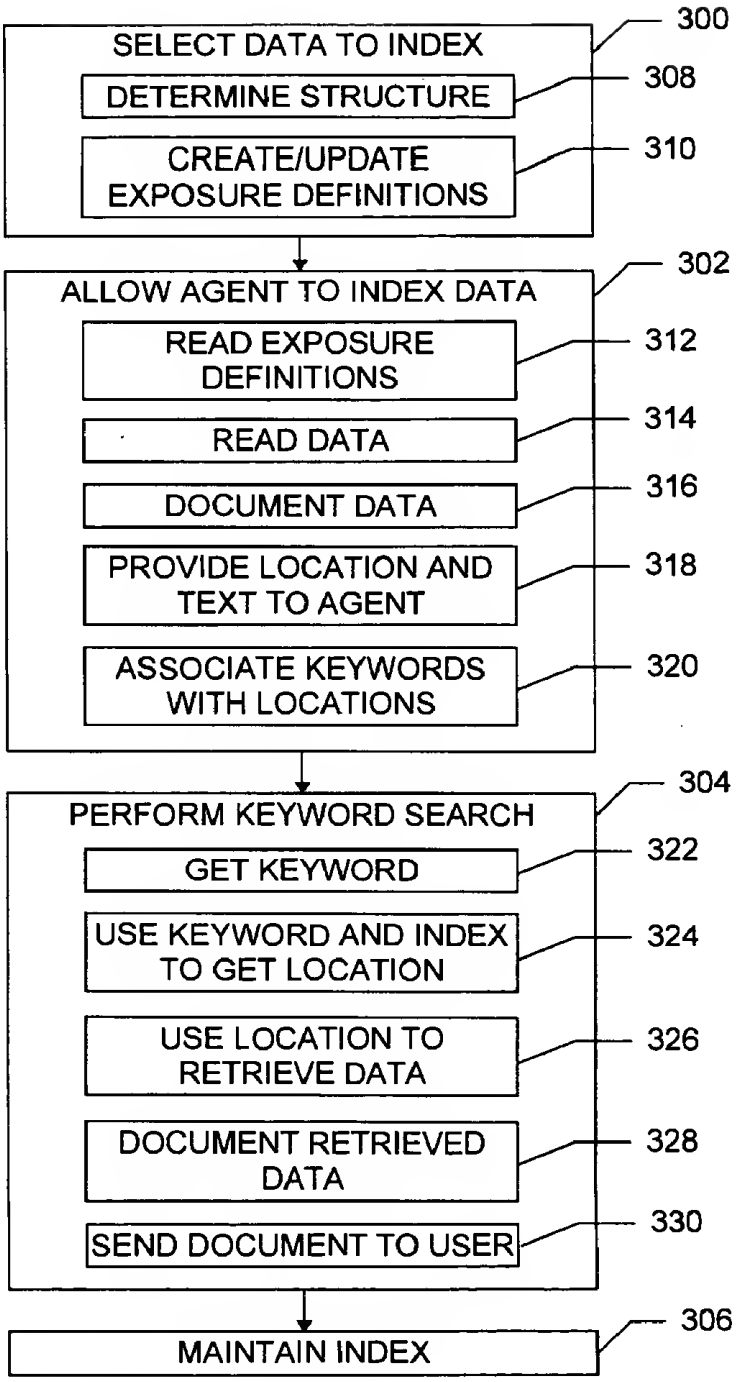

SELECT DATA TO INDEX
300
DETERMINE STRUCTURE
308
CREATE/UPDATE EXPOSURE DEFINITIONS
310
ALLOW AGENT TO INDEX DATA
302
READ EXPOSURE DEFINITIONS
312
READ DATA
314
DOCUMENT DATA
316
PROVIDE LOCATION AND TEXT TO AGENT
318
ASSOCIATE KEYWORDS WITH LOCATIONS
320
PERFORM KEYWORD SEARCH
304
GET KEYWORD
322
USE KEYWORD AND INDEX TO GET LOCATION
324
USE LOCATION TO RETRIEVE DATA
326
DOCUMENT RETRIEVED DATA
328
SEND DOCUMENT TO USER
330
MAINTAIN INDEX
306

FIG. 3

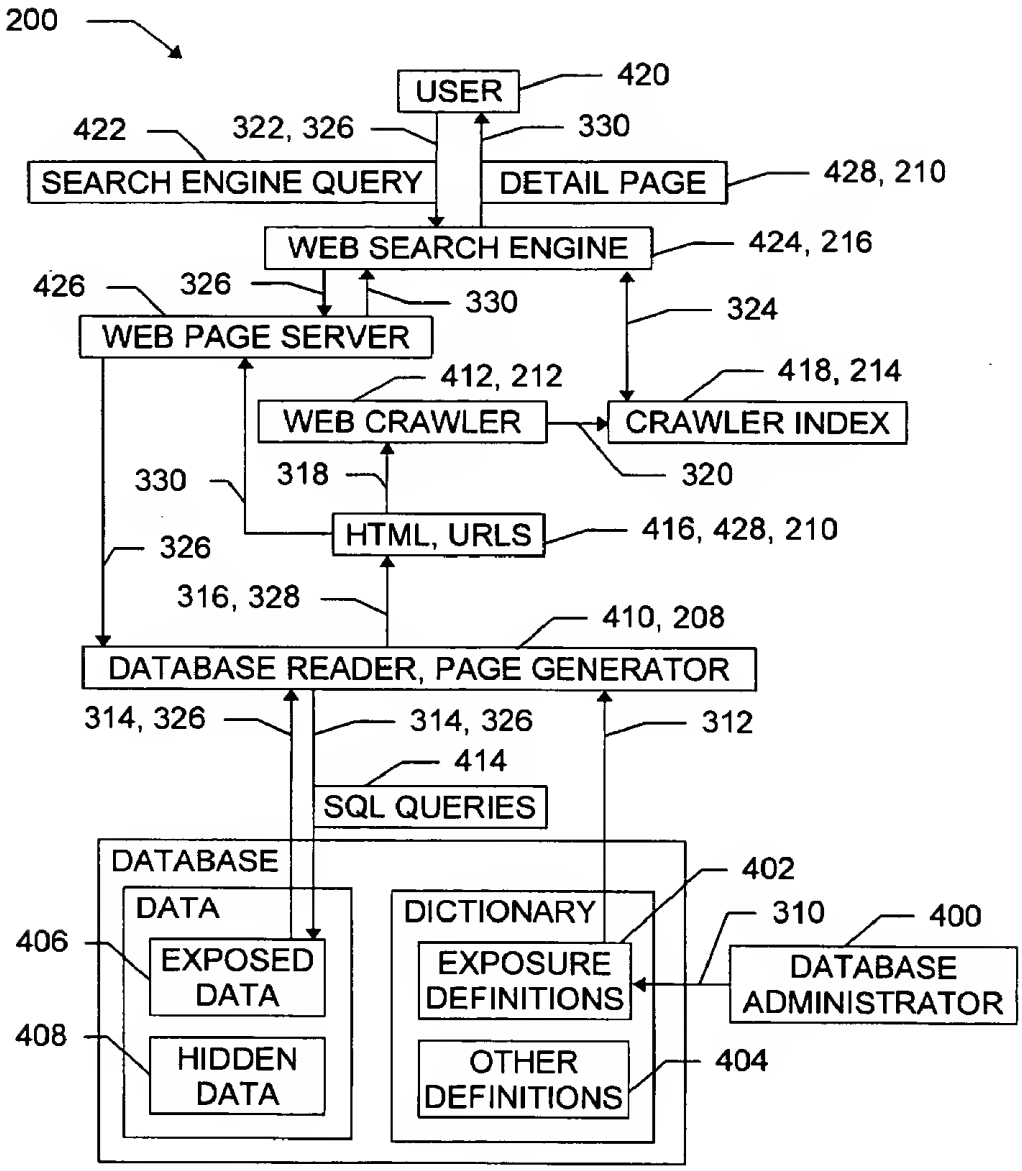

200
USER
420
422
322, 326
330
SEARCH ENGINE QUERY
DETAIL PAGE
428, 210
WEB SEARCH ENGINE
424, 216
426
326
330
324
WEB PAGE SERVER
412, 212
418, 214
WEB CRAWLER
CRAWLER INDEX
320
330
318
HTML, URLS
416, 428, 210
326
316, 328
410, 208
DATABASE READER, PAGE GENERATOR
314, 326
314, 326
312
414
SQL QUERIES
DATABASE
402
DATA
DICTIONARY
310
400
406
EXPOSED DATA
EXPOSURE DEFINITIONS
DATABASE ADMINISTRATOR
408
HIDDEN DATA
OTHER DEFINITIONS
404

FIG. 4

KEYWORD SEARCHES OF STRUCTURED DATABASES

FIELD OF THE INVENTION

The present invention relates to information management and retrieval in a digital system, and more particularly to the use of keyword indexes for retrieving data both from structured databases such as relational databases and from textual documents such as web pages.

TECHNICAL BACKGROUND OF THE INVENTION

Information is stored digitally in a wide variety of formats, which are accessed with a bewildering assortment of retrieval operations. As computers containing digital information are increasingly connected with one another, the differences between different information stores become more evident and more frustrating. Thus, many approaches have been proposed or implemented to make information more widely available.

Vast amounts of information are stored by corporations, government agencies, and other entities in structured databases, of which the most widely used are relational databases. In a typical relational database, individual pieces of data such as names, addresses, prices, and part numbers are stored in rows and columns designated by headings and organized into tables or other relations. The smallest unit of manipulation is an individual database record holding one (or perhaps a few) data values.

Indexes into the data records and tables are generated and maintained internally by database management software to make record accesses more efficient. Each database has its own set of indexes. The indexes are updated whenever a record's value is changed, or in some cases at periodic intervals. In some relational databases, all records are indexed; in others, indexes are created only after the number of records or the importance of particular records passes a threshold or another efficiency criterion is met. In many relational (and other) databases only primary database key values are indexed; other data values are retrieved by way of the keys and the relationships defined between key values and other (secondary) values. Information about the data values is provided through a database query language. The various dialects of the SQL language are among the most widely used query languages.

Enormous amounts of information are also stored in textual documents using markup languages such as HTML, XML, and other variations on SGML. Markup language document stores differ from relational databases in several important ways. The smallest unit of retrieval is typically an entire "page" (which may actually print as several pages). Each page typically contains many more words or numbers than a relational database record. The pages are not organized into tables or other relations, but are instead connected by hyperlinks or hot links. Pages may also be grouped in a file system by directory placement and/or file naming conventions.

Web crawlers and other network-roaming agents index the pages at sporadic intervals. After a given page is posted to the network, considerable time may pass before an agent encounters and indexes the page. A given index often points to information at numerous sites. The same page may be indexed in different ways by different agents. Sometimes all the words in a page are indexed, but more often selected words are indexed. Since the indexed words are selected by the web page author, they do not always impartially and accurately summarize the page's contents. The indexes are used by keyword search engines that provide users with an interface that is substantially simpler, but also less powerful, than typical SQL interfaces.

Much useful information is also stored in word processor textual documents, such as *.doc, *.pdf, *.ps, *.rtf, *.txt, and other documents. Word-processed document repositories and their associated document management systems are similar to web sites and to relational databases in some ways, and different in others. Some repositories are organized only by placing documents in particular directories in a file system hierarchy; no indexing is provided to speed searches. Other repositories index their documents according to the entire text of each document in the repository, but indexing is more commonly based on selected keywords provided by the document's author or by a human or automated subject matter classifier. Each repository has its own set of indexes. The user interface may support either a keyword search of the documents or an SQL-like query of an associated structured database of document keywords, authors, dates, titles, and similar data.

Unfortunately, the differences between these various information storage and retrieval approaches makes it difficult to provide a single interface that gives users access to information from all available digital sources. The attempts to bridge differences between different sources of information are almost as varied as the sources themselves, and fully comprehensive indexes are not available.

One approach to increasing information availability involves "dynamic HTML." An SQL query embedded in an HTML web page is extracted by a web server, sent to a relational database query handler, and processed in conventional manner by the relational database management system. The results of the query are placed in HTML format and returned to the user. This system strikes a balance between SQL's flexibility and SQL's complexity by deciding what queries are available, expressing them in natural language in the web page, and writing them in SQL ahead of time for the user. However, users who do a keyword search using a web browser or intranet search engine will not necessarily discover that the relational database contains relevant information, even if the keywords searched are among the data that would have been retrieved by the dynamic HTML query, because the web crawler index is based on the text of the dynamic HTML page, not on the relational data.

Another approach uses a natural language front-end to translate an English sentence into an SQL query which is then processed in conventional manner. The system provides greater flexibility than dynamic HTML, allowing users to write questions in a natural language and then translating the questions into SQL queries (sometimes with varying degrees of success). As with dynamic HTML, however, users who do a keyword search using a browser or search engine will not necessarily discover relevant information even if the keywords searched are among the data that would have been retrieved by an SQL query. The keyword search results might not even direct users to the natural language front-end.

Accordingly, another approach proceeds as follows. The column or table heading names and relationship names used in the database are extracted from a data dictionary that defines the relational database's structure. Selected data values are added, and then synonyms of all these terms are added, creating a list of "magnet terms." The magnet terms are placed in a web "magnet page" that also has an SQL query interface. The magnet terms will be indexed by a web crawler, so that users who do keyword searches using the

magnet terms are directed to the magnet page and its SQL query interface.

The magnet page query interface may he a dynamic HTML interface, with prewritten SQL queries accompanied hy explanatory text. The query interface may also be a natural language interface configured to receive English questions and translate them into SQL queries. Or the query interface may simply accept SQL queries and pass them to the database management software. Of course, the query interface may also combine dynamic HTML, natural language translation, and straightforward SQL querying capabilities.

In any case, a SQL query from the query interface is directed to the relational database, which uses its internal indexes to retrieve tbe data. The results are packaged as HTML and displayed to the user. This approach has the advantage tbat if tbeir keywords arc among the magnet terms, tben users wbo do a keyword search will he directed to the magnet page for the relational database containing the relevant information. However, users will usually not reach tbe query interface unless the data they seek appears in the magnet terms. Moreover, even if they do reach the query interface they must still find or formulate an SQL query that will retrieve tbe relevant information from the database.

Instead of attempting to make relational database information available to web browsers, a different approach tries to make web pages accessible througb a relational database interface. Text documents such as plain text files, HTML pages, word processor documents, and the like are entered as records in a relational database. Keywords or the full text of the documents are entered in tbe database's internal indexes to support document retrieval tbrough the database query interface using SQL or another query language.

This approach has the advantage of bringing powerful and well-understood relational database software to hear on the problem of retrieving relevant text documents. But users who browse a network on wbicb the relational database occupies only one or a few nodes will not necessarily realize that tbc information they seek rcsides in documents indexed into the datahase in question, even if tbe keywords they use in their browsing appear in the document indexes. The indexes are internal to tbe database and tbus are used only in response to SQL or like queries directed specifically at the database.

Otber approacbes are also described in the literature and/or embodied in software currently being used. For instance, structured databases other tban relational databases are sometimes used, including hierarchical, object-relational, object-oriented, and other structured databases. Also, at least one web crawler now indexes word processor documents as well as markup language documents. But the examples above illustrate several important characteristics of different approaches to puhlishing information:

- the smallest unit of data retrieved (e.g., database record, web page);
- the rules used to organize data (e.g., relations, file placement and naming conventions, hyperlinks);
- how data is retrieved (e.g., SQL queries, keyword searches);
- what data is indexed for each data unit (e.g., headings, primary database keys, author-defined keywords, selected keywords, full text);
- wbere the indexes reside (e.g., witbin tbe database system or outside it);
- wbich sources are indexed (e.g., tbe records of a given database, the web sites visited by the crawler); and
- when the index is updated (e.g., when the record is entered or modified, periodically, when tbe crawler visits the site).

Wben existing approaches are viewed in the manner discussed ahove, it hecomes apparent that improvements are possible. For instance, it would be an advancement in the art to make structured database information visible to net-wide keyword searcbes wben a user bas not yet identified tbe database in question as one likely to contain relevant information.

It would be an additional advancement to provide such a method and system which do not interfere with existing retrieval mecbanisms, but serve instead as additional tools for identifying and retrieving information based on keywords.

Sucb a metbod and system are disclosed and claimed herein.

BRIEF SUMMARY OF THE INVENTION

The present invention provides a method and system for supporting keyword searches of data items in a structured database, such as a relational database. One metbod of tbe invention bcgins with selection of at least one data item in the structured datahase; each selected item contains data and has a corresponding location identifier which identifies the item's location within tbe structured database. For instance, a relational database record may be identified by an object class name and one or more primary database key values.

The selected data items are documented by creating at least one document, sucb as a web page, wbicb resides outside tbe structured database as a memory stream or as a file and which contains a textual representation of eacb selected item's data. The documents are tben indexed by creating an index outside the database which associates keywords in tbe textual representation of eacb selected item's data witb that item's location identifier. The indexed keywords are more comprehensive and accurate than terms used in conventional magnet pages or web page meta content tags because they are generated directly from most or all of tbe data values.

If tbe structured datahase includes data items organized as records in relations according to a data dictionary, then selection may be accomplished by providing a supplemental data dictionary whicb identifies the selected records or tables. In this case, tbe indexing step only indexes records and tables that are identified by the supplemental data dictionary. A data dictionary may also be used to identify selected data items for binary-only relational databases that have no accessible data dictionary and for non-relational databases.

Indexing may be accomplished by providing to a keyword search engine indexing agent both the textual representation of each selected item's data and the selected item's location identifier. The indexing agent produces an index that associates keywords with resource locators, and eacb resource locator includcs a textual rcpresentation of a data itcm location identifier. Suitable indexing agents include web crawlers, indexing "bots", and other text indexing tools. Suitable resource locators include URLs, bot links, file paths, and distinguished names, object class names, table names, and primary database key values, among otbers.

Users provide keywords to a searcb engine interface in a system according to the invention. The system uses tbe index to obtain a resource locator that is associated with the keyword. The resource locator is used to retrieve the item's current data from the structured database, using SQL queries

or other structured database retrieval mechanisms. A document containing the retrieved data, such as a web page, is then generated and provided to the user.

The invention bridges a gap between loosely structured textual keyword search information technologies, on the one hand, and highly structured relational/hierarchical query language search database technologies, on the other. Web pages on the Internet or on an intranet are effective for textual information that is relatively static and unstructured, such as press releases, user guides, policy statements, and procedure manuals. Other information, such as availability, pricing, performance and planning records, is more dynamic and has traditionally been maintained in highly structured databases such as relational or object-oriented databases.

The invention makes it possible to use a single search method—keyword searching—to locate and retrieve desired information from different types of information sources. In particular, the invention makes it possible to publish selected portions of a relational database in a manner that allows users to retrieve relational data without knowing details of the database's internal organization. Other features and advantages of the present invention will become more fully apparent through the following description.

BRIEF DESCRIPTION OF THE DRAWINGS

To illustrate the manner in which the advantages and features of the invention are obtained, a more particular description of the invention will be given with reference to the attached drawings. These drawings only illustrate selected aspects of the invention and thus do not limit the invention's scope. In the drawings:

FIG. 1 is a diagram illustrating one of many networks suitable for use according to the present invention.

FIG. 2 is a block diagram further illustrating components of the network shown in FIG. 1 and other suitable systems according to the invention.

FIG. 3 is a flowchart illustrating methods of the present invention.

FIG. 4 is a data flow diagram illustrating components and methods of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a method and system for assisting keyword searches of highly structured data. Before detailing the architecture of methods and systems according to the invention, the meaning of several important terms is clarified. Specific examples are given to illustrate aspects of the invention, but those of skill in the art will understand that other examples may also fall within the meaning of the terms used. Some terms are also defined, either explicitly or implicitly, elsewhere herein.

Terminology

As used here, a "keyword" search is a pattern-matching search which tries to locate instances of digital data using a key word or phrase. Many conventional web search engines support keyword searches. Keywords may contain wildcards. For instance, if the question mark is used as a wildcard capable of matching any single character and the asterisk is used as a wildcard capable of matching any zero or more characters, then the keyword "b?t*" would match the words "bat", "bet", "bit", "bot", "but", "battle", "bitten", and "butane", among others. In some cases keywords may also contain regular expressions, such as the regular expressions used in the familiar lexical analysis program lex or the familiar text editors emacs and vi. A keyword may contain smaller keywords connected by operators such as AND and OR.

One alternative to keyword searching is "browsing" through the available data until values of interest are located. Browsing is available in most computer information management systems, regardless of whether keyword searches are supported. An important difference between keyword searching and browsing is that keyword searches focus much more quickly on portions of the data that are likely to be of interest. This is particularly true if the keyword search is performed on data that is grouped by subject matter. For instance, a search using the keyword "bat" in data classified by subject matter could lead quickly to baseball statistics rather than a discussion of flying mammals.

Many conventional structured database systems support "query" searches through SQL or another query language. An important difference between query searches and keyword searches is that query searches normally presume the existence of relations or other structure in the data and contain assumptions about that structure For instance, many SQL queries are of the form SELECT X FROM Y WHERE Z, with X being the heading name of a column in a table called Y, and Z being some constraint on the values stored in the column. Such a query will be rejected if no table named Y exists, or if Y exists but has no column named X.

By contrast, keyword searches typically assume nothing about the relationships or structures that may internally connect different instances of matching data. In particular, a keyword search of a relational database according to one embodiment of the present invention for a keyword K will identify all data values in the exposed portion of the database that match K, regardless of the table names or column names being used.

Even if a particular relational database system supported queries such as SELECT ALL FROM ALL WHERE (ENTRY CONTAINS 'K'), this would not be equivalent to a system according to the invention which assists a keyword search of all database records for matches to the keyword K. For instance, the internal indexing and retrieval mechanisms in relational databases are optimized for selecting and combining records in rows and columns and tables according to the database structure as well as testing data value constraints; these mechanisms are not optimized for retrieving every data value and then testing it against the key. Also, web crawlers and other keyword index builders index all data values supplied to them, while relational databases typically index only selected columns or rows. Finally, indexes according to the invention will generally have a much broader context or scope than an internal relational database index, involving not just a single relational database but many other information sources as well; this makes the inventive indexes more useful with all-purpose or comprehensive search efforts.

As used here, a "structured database" is a collection of data items organized primarily by rules other than those governing natural languages such as English. The data items may contain natural language text such as addresses or part names in a relational database, but relations, tables, trees, or other structures are the primary means of organization. Structured database operations aid decision-making by allowing users to combine individual data items in various ways, as illustrated in the SQL query above.

Relational databases are one example of structured databases; other examples include hierarchical, inverted-list, object-relational, object-oriented, and flat-file databases. Structured databases may be stored in a single location or

distributed between several machines. Regardless of the approach taken to storage, many structured databases can be accessed through a network.

As used here, "network" includes local area networks, wide area networks, metropolitan area networks, and/or various "Internet" networks such as the World Wide Web, a private Internet, a secure Internet, a value-added network, a virtual private network, an extranet, or an intranet. One of many possible networks suitable for use according to the invention is shown in FIG. 1, as indicated by the arrow labeled **100**. The network **100** includes a server **102** and several clients **104**; other suitable networks may contain other combinations of servers, clients, and/or peer-to-peer nodes, and a given computer may function both as a client and as a server. The computers connected by a suitable network may be workstations, laptop computers, disconnectable mobile computers, servers, mainframes, so-called "network computers" or "lean clients", personal digital assistants, or a combination thereof.

The network may include communications or networking software such as the software available from Novell, Microsoft, Artisoft, and other vendors, and may operate using TCP/IP, SPX, IPX, and other protocols over twisted pair, coaxial, or optical fiber cables, telephone lines, satellites, microwave relays, modulated AC power lines, and/or other data transmission "wires" known to those of skill in the art. The network may encompass smaller networks and/or be connectable to other networks through a gateway or similar mechanism.

As suggested by FIG. 1, at least one of the computers is capable of using a floppy drive, tape drive, optical drive, magneto-optical drive, or other means to read a storage medium **106**. A suitable storage medium **106** includes a magnetic, optical, or other computer-readable storage device having a specific physical configuration. Suitable storage devices include floppy disks, hard disks, tape, CD-ROMs, PROMs, random access memory, and other computer system storage devices. The physical configuration represents data and instructions which cause the computer system to operate in a specific and predefined manner as described herein. Thus, the medium **106** tangibly embodies a program, functions, and/or instructions that are executable by computer(s) to assist keyword searches of structured data substantially as described herein.

Suitable software for implementing the invention is readily provided by those of skill in the art using the teachings presented here and programming languages and tools such as Java, Pascal, C++, C, CGI, Perl, SQL, APIs, SDKs, assembly, firmware, microcode, and/or other languages and tools.

Overview of Components

An overview of the main components of the invention and its environment is now given with reference to FIG. **2**. A system **200** according to the invention operates using the network **100** or another suitable computer system. A structured database **202** and corresponding exposure definitions **204** are part of the inventive system or accessible to the inventive system **200**. The structured database **202** includes data items which have data values; suitable databases include conventional relational databases and other conventional structured databases with the associated database management system software.

The exposure definitions **204** identify the portion(s) of the structured database **202** that will be exposed to external keyword searches; the entire database **202** is typically already searchable by SQL or other conventional query means. Those of skill will appreciate that the system **200** can also be configured such that the exposure definitions **204** identify the portions of the database **202** which should NOT be exposed for keyword searching, if that approach is more efficient or convenient. In either case, the exposure definitions **204** may be in the form of a data dictionary, particularly if the structured database **202** is a relational database. However, the exposure definitions **204** may also take the form of a schema, particularly if the structured database **202** is a hierarchical database or other database defined by a schema.

In the illustrated system **200**, the exposure definitions **204** are created and edited using an administration tool **206**. The tool **206** may operate by extracting the definitions **204** from an existing data dictionary or schema, or it may be necessary to build the definitions from scratch by reverse engineering the data formats used in a binary-only structured database **202** and then generating a data dictionary or schema which can be edited to eliminate portions of the database **202** that should not be exposed.

A document generator **208** generates documents **210** which contain textual representations of the exposed data values in the database **202**. In one embodiment, the document generator **208** generates a document, such as an HTML page, for each table in a relational database **202**, containing the table's values in ASCII form, and then locates the document **210** at a Uniform Resource Locator (URL) corresponding to the table's location in the database **202**. For instance, an HTML page containing the data values stored in a sales database table named "customers" might be generated and then stored at http://www.company.com/salesdb/customers.htm.

An indexing agent **212** reads the documents **210** and generates entries in an index **214**. Suitable indexing agents **212** include web crawlers, spiders, indexing robots, and other indexing tools. The indexing agent **212** may be a network-roaming agent, or it may be tied to one or a few network sites. In one embodiment of the system **200**, the indexing agent **212** indexes every data value in each document **210**, not just "meta tag" or other values that may or may not be representative of the actual database contents. Unlike indexing processes running inside the structured database **202**, the indexing agent **212** does not rely heavily on assumptions about the database structure but merely treats the documents **210** as sources of text which have little or no structure except that imposed by English or another natural language.

A keyword search engine user interface **216** may be integral with the indexing agent **212**, or it may be a separate program provided by a separate vendor. The user interface **216** accepts keywords (possibly including wildcards) and uses the index **214** and possibly other components of the system **200** to locate corresponding documents **210**.

Overview of Operation

An overview of the operation of the system **200** is now given, with reference to FIGS. **2** and **3**. Four main steps are shown in FIG. **3**: a data selecting step **300**, an index allowing step **302**, a search performing step **304**, and an index maintaining step **306**. These steps may be grouped for ease of explanation into an indexing phase (steps **300**, **302**, and **306**) and a searching phase (step **304**). During the indexing phase, the index **214** is created or updated. During the searching phase, the index **214** is used to respond to keyword searches directed at the database **202** (and often to other information sources as well). In practice, both phases may be happening simultaneously or in an interleaved fashion.

The selecting step **300** illustrated includes a structure determining step **308** and a definition editing step **310**.

During the determining step **310**, the administration tool **206** determines what structures are being used in the structured database **202**. For instance, the tool **206** may read an existing data dictionary (sometimes called a "catalog") of a relational database **202** or an existing schema for a hierarchical or object-oriented database **202** and then identify the relations, partitions, record types, data types, links, indexes, primary database keys, and other structures used to organize the database **202**. If no data catalog or schema exists, the tool **206** may be used to assist one of skill in reverse engineering the structure definitions by examining the binary contents of the database **202** together with display formats, documentation, and any other available structural information.

During the editing step **310**, the exposure definitions **204** are initially created and/or updated by the tool **206**. Some embodiments favor ease of editing by closely modeling the exposure definitions **204** after an existing data dictionary or schema for each database **202**, while others favor portability in the document generator **208** by making all exposure definitions **204** for all databases **202** use a common format, such as a particular relational database data dictionary format.

In any case, the selecting step **300** selects at least one data item in the structured database **202**, with each selected item containing data and each selected item having a corresponding location identifier which identifies the item's location within the structured database **202**. Suitable location identifiers include table, row, and/or column names; unique relational data key values; paths, filenames, common names, contexts, and/or distinguished names; offsets, pointers, and/or record numbers; pointer array or hash table indexes or entry numbers; transaction numbers or sequence numbers; universal unique identifiers (UUIDs) or globally unique identifiers (GUIDs); and combinations of such identifiers. The name or location of the database **202** may be part of a suitable location identifier, but merely identifying the database **202** is not sufficient.

The allowing step **302** illustrated includes a definition reading step **312**, a data reading step **314**, a data documenting step **316**, a providing step **318**, and an associating step **320**. During the definition reading step **312**, the document generator **208** reads the exposure definitions **204** and builds or locates a checklist that will be used to make sure all selected data is exposed for indexing.

During the data reading step **314** the document generator **208** reads the selected data from the database **202**. Data reads may be performed directly from the binary database **202** using low-level file system commands, but it may be better to retrieve the data using the using the SQL interface, application program interface (API), or other existing data retrieval software of the database **202**. Data reads may be done all at once, but more often the data reading step **314** and the data documenting step **316** will be repeated in pairs, so that a chunk of data is read and then documented, the next chunk of data is read and documented, and so forth until all selected data is documented. Of course, the providing step **318** and the associating step **320** may also be made part of the loop, so that each chunk of data is indexed before the next chunk is read.

More generally, FIG. **3** shows a particular order and grouping for the main steps **300** through **306** and for various subsidiary steps. However, those of skill in the art will appreciate that the steps illustrated and discussed here may be performed in various orders, except in those cases in which the results of one step are required as input to another step. Likewise, steps may be omitted unless called for in the claims, regardless of whether they are expressly described as optional in this Detailed Description. Steps may also be repeated, or combined, or named differently. In one alternative embodiment, for instance, an "indexing" step includes the step **318** of providing to the keyword search engine indexing agent **212** both the textual representation of each selected item's data and the selected item's location identifier.

During the data documenting step **316**, the document generator **208** documents the selected data items by creating at least one document outside the structured database **202**; the document(s) **210** contain a textual representation of each selected item's data. The document may exist as a stream of data in RAM or coming from a network or other connection. The document may also be stored on disk as a file, but those of skill will appreciate that throughput generally increases when disk accesses are reduced or eliminated. An index such as the index **214**, a web crawler index, or an internal database **202** index, is not a suitable result of the documenting step **316**. Rather, textual documents produced by the step **316** include plain text or word processor documents, as well as markup language documents.

Markup language documents use markup language formats such as Standard Generalized Mark-up Language (SGML), which is specified in the 1986 International Standards Organization Standard No. 8879. Familiar markup languages include HTML and XML. Other mark-up languages are used in Folio infobases, Microsoft Word documents, Corel WordPerfect documents, troff documents, and various hyperlink and hypertext documents (MICROSOFT WORD and COREL WORDPERFECT are marks of Microsoft and Corel, respectively). Mark-up languages generally provide links which associate a particular, pre-selected location in a primary text file with additional text, images, or other information, or with links to email, display, or other software.

In one embodiment, documents **210** produced with the step **316** include a comprehensive textual representation of each selected item's data. "Comprehensive" means that every data value, or at least substantially every data value, appears separately in the documents **210**. Every exposed data value that might reasonably be used as a keyword should appear in the documents **210**. Merely listing table, row, column, partition, subtree, or other group names is not sufficient, although these may be treated as data values and placed in the documents **210**. Nor is it adequate to summarize data or to select a relatively small sampling of "representative" or "boundary" or "central" data values.

However, common terms such as "a", "the", "not" and the like may be omitted from a comprehensive representation of data values to conserve space and improve keyword search efficiency. Also, comprehensiveness may be with respect to all selected (exposed) data values, or merely with respect to non-numeric exposed data values or some other efficiency grouping. For instance, a comprehensive index may include all selected data values for part numbers and customer names but exclude prices and dates in the selected data items.

During the providing step **318**, the location of selected data in the database **202** and the textual representation of the selected data's values are provided to the indexing agent **212**. If the agent **212** is a roaming agent, such as a web crawler, this may be accomplished by storing the documents **210** in files having names that contain the database locations of the documented data and then making the files accessible for indexing by the crawler. For instance, an HTML document **210** containing the textual representation of data values

stored in a database **202** table named "customers" could be stored in a file named "customers.htm", or an XML document **210** containing the textual representation of data stored in an object database **202** could be stored in a file whose path name includes a class identifier, file type, and GUID, such as "/OLE/dll/42754580-16b7-11ce-80eb-00aa003d7352". If the agent **212** does not roam the system **200**, then steps must be taken to bring the agent **212** together with the paired locations and textual data, such as by providing the pairs directly or indirectly as command line parameters or as interactive input to the agent **212**.

During the associating step **320**, the agent **212** associates the textual data values with their paired location(s) in the index **214**, treating the data values as keywords. That is, the associating step **320** indexes the documents **210** by creating or updating the index **214** (which resides outside the database **202**) so that the index **214** associates keywords in the textual representation of each selected item's data with that item's location identifier.

The index **214** and the indexing agent **212** may use B-trees, hashing, and other familiar data structures and operations to create or modify or extend the index **214**. If the documents **210** are in HTML format and the agent **212** is a web crawler that only indexes meta content tag values then comprehensive indexing places all (or substantially all) data values in the meta content tags so they will be indexed by the agent **212**.

In one embodiment, the agent **212** produces an index **214** that associates keywords with resource locators, and each resource locator includes a textual representation of a data item location identifier. Suitable resource locators include URLs (including hot links), file names, file path names, GUIDs, distinguished names, database key values, object or class or table or column names, and other resource identifiers.

A major advantage of the present invention is that the index **214** will tend to contain entries for data sources other than the database **202**, unlike the internal database **202** indexes. For instance, the index **214** may associate keywords with storage locations in multiple relational and other databases, web sites, file systems, word processor document management systems, Lotus Notes (mark of IBM) databases, Microsoft Exchange (mark of Microsoft) databases, and other data sources.

Moreover, adding structured database **202** values to an existing index **214** with the invention leverages the existing values in the index **214**, the existing indexing capability of the agent **212**, existing search engine interfaces **216**, and existing document **210** formats. The invention extends these capabilities, rather than attempting to replace them by forcing use of yet another closed, proprietary data format.

The keyword search performing step **304** illustrated includes a keyword obtaining step **322**, an index using step **324**, a retrieving step **326**, a documenting step **328**, and a transmitting step **330**. During the keyword obtaining step **322**, the user interface **216** obtains a keyword from a user. The user may be a human, or it may be a task, thread, or other computer process. The keyword may be a single word, a portion of a word with one or more with wildcards, or a combination of such words. Combinations are formed using familiar text search operators such as And, Or, But Not, Within N Words, Within Same Sentence, and the like. Keyword searches may be performed in the context of subject matter, chronological, or field scope constraints.

During the index using step **324**, the search engine **216** uses the index **214** to obtain the location(s) of instances that match the keyword. Although an integrated interface and search engine **216** is illustrated, in other embodiments the index-using search engine is separate from the user interface and may even accept keyword searches from several different user interfaces. Familiar pattern-matching and lookup techniques, such as those currently available through Yahoo!, Digital Alta Vista, Infoseek, and Excite web sites (marks of their respective owners) and other keyword search engines may be used during the step **324**.

During the retrieving step **326**, documents **210** containing instances of the keyword may be supplied to the search engine **216** for transmission to the user; no is documents are supplied if no matches are found. The documents **210** may have been created during the documenting step **316** as part of the indexing phase, or they may be created in response to the keyword search being performed during the step **304**.

In the latter case, the search engine **216** and the document generator **208** use the location information obtained from the index **214** to retrieve data values from the structured database **202** and then create corresponding documents **210** during the step **328**. In one embodiment, only the individual data values that match the keyword and reside in the selected data items are retrieved. In another embodiment contextual information, such as nearby data values or table names, is also retrieved and documented. Retrieval during the step **326** may otherwise proceed generally as discussed in connection with the data reading step **314** above. The documenting step **328** may proceed generally as discussed in connection with the documenting step **316** above.

The step **330** may send documents **210** to the user interface **216** to be displayed on a screen as part of a graphical user interface, stored in a file, or otherwise used. The documents **210** may be summarized, compressed, encrypted, translated, or otherwise manipulated before, during, or after their transmittal.

The index maintaining step **306** proceeds generally like the allowing step **302**, except that only some of the selected data items are indexed. For instance, a log of changes to the structured database **202** may be maintained by the database **202** or by the administration tool **206**, so that only data values that may have changed are re-indexed.

Additional Examples

FIG. **4** illustrates further the components, environment, and operation of one embodiment of the invention; reference is also made to the earlier figures. FIG. **4** provides one of many possible examples; steps and/or components may be added, omitted, re-ordered, and/or performed concurrently in other embodiments according to the invention.

During the indexing phase, a database administrator **400** performs the editing step **310** by using the administration tool **206** to create exposure definitions **204** in the form of data dictionary definitions **402**. A pre-existing data dictionary **404** defines the structure of the entire database **202**; the exposure definitions **204** divide the data into a portion **406** which is exposed for indexing and a portion **408** which will not be indexed into the index **214**. The data dictionary **402** may also be used to associate selected classes with specific tables or views, to associate default named attributes and attribute types with each selected table column, and to assist operations such as data type conversion and output formatting.

During the definition reading step **312**, a combination database reader and page generator **410** (which act as the document generator **208**) reads the data dictionary **402** to identify the portion of the database **202** that will be exposed to a web crawler **412** (which acts as the indexing agent **212**). If the administrator **400** wishes to create a virtual record that is the join of several tables so that users **420** receive

additional context in search results, the administrator **400** can use the tool **206** and the dictionary **402** to do so, and the database reader **410** will treat the resultant join as a composite record.

During the data retrieving step **314**, the database reader **410** creates SQL queries **414** which will extract the exposed data **406**, queries the database **202**, and buffers the extracted data **406**. During the documenting step **316**, the page generator **410** creates HTML pages **416** containing the extracted data **406**. The URL associated with each HTML page **416** includes a textual representation of the location in the database **202** from which the data represented in the page **416** was extracted.

During the providing step **318**, the HTML pages **416** are made accessible to one or more web crawlers **412**, along with the corresponding URLs generated by the page generator **410**. During the associating step **320**, the web crawler **412** reads the HTML pages **416** and creates or updates an index **418**. This concludes the indexing phase, or at least the first iteration of the indexing phase; subsequent indexing may be interleaved with keyword searches or performed concurrently with such searches.

In the search phase, during the keyword obtaining step **322** a user **420** enters a keyword search **422** into a web or Internet or intranet search engine **424**. During the step **324**, the search engine **424** uses the crawler index **418** to generate search results that (for purposes of illustration we will assume) contain URLs generated by the page generator **410**. During one version of the retrieving step **326**, the corresponding pages **416**, which were generated during the indexing phase, are then supplied to the search engine **424** for transmittal to the user **420**. The search phase may end at this point.

However, during another version of the retrieving step **326**, the user **420** may also request (implicitly or expressly) additional detail about a keyword search result whose URL was generated by the page generator **410**, or the most current possible results. In response, the search engine **424** asks a web page server **426** for the HTML page located at the URL. The web server **426** asks the database reader **410** for the HTML page. The database reader **410** uses the data dictionary **402** to formulate a SQL query **414** for the corresponding current data, based on the data location information embedded in the URL. The database reader **410** accepts the SQL query response and buffers it. During the step **328**, the page generator **410** creates detail HTML pages **428** containing the current data provided in the SQL query response. Finally, during the transmitting step **330**, the page generator **410** makes the detail HTML pages **428** accessible to the web page server **426**, which passes the detail HTML pages **428** to the search engine **424**, which displays the detail HTML pages **428** to the user **420**.

In one alternative embodiment, the structured database **202** includes data items organized as records in relations according to the data dictionary **404**, the selecting step **300** includes the step of providing the supplemental data dictionary **402** which identifies selected records or tables, and the indexing step **320** only indexes records and tables that are identified by the supplemental data dictionary **402**.

In some embodiments, the computer system **200** includes a selecting means for selecting data items in the structured database **202**. Suitable selecting means include the exposure definitions **204** and/or **402**, an exposure definition schema defining exposed elements of the database **202**, the administration tool **206**, software and/or hardware implementing the selecting step **300**, and/or other selecting means, in appropriate combinations.

In some embodiments, but particularly if the structured database **202** includes a relational database and the data items include relational database records or tables, the selecting means includes the selection data dictionary **402** which specifies only selected relational database records or tables. The data dictionary **402** may be used when other definitions **404** are present, or when they are not, and may be used even if the database **202** is not entirely relational.

The system **200** also includes a retrieving means for retrieving from the database **202** the current data of a selected data item, such as the document generator **208**, search engine **424**, database reader **410**, document server **426**, software and/or hardware implementing the retrieving step **326**, and/or other retrieving means, in appropriate combinations.

In addition, the system **200** includes an exposing means for exposing to the indexing agent **212** information about a data item's location in the database **202** together with information about the data item's retrieved data. Suitable exposing means include the document generator **208**, page generator **410**, documents **210** and/or **416** and/or **428**, software and/or hardware implementing the documenting step **316** or providing step **318**, means for invoking the agent **212** or crawler **412**, and/or other exposing or documenting means, in appropriate combinations.

In one embodiment, the search engine interface **216** and the retrieving means reside on different nodes in the network **100** and communicate with one another using a TCP/IP network protocol. In another embodiment, communication is accomplished using an IPX network protocol.

In one embodiment, the administration tool **206** and other system **200** components are compatible with widely used commercial operating system, networking, and database management software and systems, and include a user interface designed to prevent confusion by limiting administrator **400** access to one set of exposure definitions **204** at a time. For instance, one embodiment supports the data dictionary **404** table layouts for major commercial database vendors such as Oracle, SQL Server, Sybase, and Informix. Different database vendors may have different names for different data types, so all types in the data dictionary **404** are coerced into one of the following types: Date; Number (includes at least Integer, Real, Float); and Char (includes at least VarChar2, Long).

At least initially, implementation may be eased by not supporting RAW or BLOB data types, but support for these and other types is included in alternative embodiments of the invention. Likewise, both textual and relational/structured information stores are becoming better adapted for use with graphical and audible data, such as static images, video clips, and audio files. Terms such as "textual" and "data value" used herein should be understood to include such digital forms of multi-media and audiovisual information.

The capabilities available through this embodiment of the tool **206** in an "Admin" menu include: New (start new exposure definitions **204**); Open (open existing set of exposure definitions **204** for review and possible editing or copying); Save or Save As (save exposure definitions **204** in a file); Project (edit configuration values such as database **202** name, database **202** user ID and password); Generate (generate an HTML index file and HTML template files for each object class in the target directory for a currently open set of exposure definitions **204**); Initialize (drop and create database dictionary tables in the current database **202** account); and Exit.

In this embodiment, information needed to connect the tool **206** to the database **202** includes: a file name (full path)

for the exposure definitions **402** and other configuration values; directory location(s) for HTML output template files; a database name (displayed at top of every output HTML page **210** in case multiple databases are crawled and indexed together); and a database user ID, password, and connection string (used by the tool **206** and the database reader **410** to log into and read the database **202**). In one alternative embodiment, the information provided to the tool **206** also includes a directory location for an HTML index file **214**.

The capabilities available through this embodiment of the tool **206** in an "Objects" menu include: Object Screen (list of database **202** user names populated on entry leads to list showing tables and views owned by selected user and object class information defined for each table); Attribute Sub-Screen (column names for table are queried and displayed; for newly defined objects with no existing attribute records, the column names are inserted in data dictionary first and then queried; by default, attributes are populated such that attribute labels are same as column name, sequence is same as column sequence, display flag is on, primary key flag is off, character data types are given an HTML string tag and domain Text, number data types are given an HTML numeric tag and domain Number (9999), and no units are initially assigned); Object Detail Sub-Screen (object details queried and displayed on entry; new object details may be defined by selecting from a list of currently defined object classes); Object Detail Attributes Sub-Screen (defines attributes for object detail, similarly to Attribute Sub-Screen, except that join conditions between object detail and object class must be defined, as by selecting attributes from lists in current object class and object detail).

The capabilities available through this embodiment of the tool **206** in a "Domains" menu include a Domain Screen. On entry, a list is populated with the domain names currently defined. As a domain is selected, the field values are displayed. The administrator **400** can add, update, and delete domain field values. By default, the following domains should be defined on creation of a data dictionary **402**: Text (tagged as a key identifier), Text (plain), Number (9999), Number (9,999), Money ($9.99), Money ($9), Percent (9%), Percent (9.9%), Percent (9.99%), Date (MM/DD/YY), Date (DD-MON-YY).

The capabilities available through this embodiment of the tool **206** in a "Units" menu include a Units Screen. On entry a list is populated with the unit types currently defined. As a unit type is selected, the fields are displayed along with related units child records. The administrator **400** can add, update, and delete unit field values.

In one embodiment, the database reader **410** includes a crawler interface and the system **200** operates as follows. The crawler **412** crawls an URL for an index page **416** containing a list of hot links to all selected object classes. As the crawler follows the link from the index page **416** for each object class, the database reader **410** retrieves the corresponding record from the database **202** and feeds matching HTML text to the crawler **412** for indexing. HTML pages representing retrieved data are generated by the page generator **410**.

The crawler **412** can work in two modes. In a Full Scan Mode, all selected records of the table are crawled and indexed. In an Update Only Mode, only records which have been added, updated, or deleted are retrieved and crawled. Updated records can be identified by logging them in a transaction table for the object class with their primary database key and a timestamp. The log must be updated as logged records are crawled. Transaction table columns include the primary key column(s) of the object class, an action code column (Add, Update, or Delete), and a timestamp column.

In one embodiment, the database reader **410** includes a query interface and the system **200** operates as follows. After the user **420** queries records in the crawler index **418**, the user **420** seeks the current detailed database record. After selection of the hot link to the record, the database reader **410** queries the target table according to the location parameters in the hot link, which are the object class name and the primary database key values. The database reader **410** buffers the record and invokes the page generator **410**, and the HTML text is sent back to the user **420** as previously described.

In addition, the following capabilities are provided in some embodiments of the database reader **410**. Column level stored functions are defined at the domain or attribute level which allow the value of a database **202** column to be modified at query time. Input parameters for a domain level stored function include the column value and domain ID, and input parameters for an attribute level stored function include the column value, attribute ID, and row ID of the database **202** record. An output format mask is provided for numeric and date column data types. Unit scale conversions are supported. Multicolumn primary database keys for object classes and object details are supported. Finally, support is provided for managing multiple object classes and their detail records which are children of a parent object class record.

In one embodiment, the page generator **410** operates such that all database **202** column output is converted to ASCII or another character format and displayed according to the HTML template page for the particular object class involved. The format specification for template fields is in the form <object_class_name>.<attribute_label>. The name format for HTML template files is <object_class_table name>_tmplt.htm. Object class and database **202** name are displayed at the top of the generated page **416**. Field alignment is center, right, or left, with left justification being the default.

In summary, the present invention provides a novel system and method for making structured database contents available through keyword searches. By making it possible to use web crawler indexes to locate relational database records and object-oriented database objects as well as word processed documents and web pages, the invention reduces the complexity and inefficiency of searches spanning heterogeneous data sources. Moreover, the invention leverages existing information and technology resources instead of requiring users to adopt expensive new systems that are not compatible with existing resources.

Although particular methods embodying the present invention are expressly illustrated and described herein, it will be appreciated that apparatus and article embodiments may be formed according to methods of the present invention. Unless otherwise expressly indicated, the description herein of methods of the present invention therefore extends to corresponding apparatus and articles, and the description of apparatus and articles of the present invention extends likewise to corresponding methods.

The invention may be embodied in other specific forms without departing from its essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. Any explanations provided herein of the scientific principles employed in the present invention are illustrative only. The scope of the invention is, therefore, indicated by the appended claims

rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

What is claimed and desired to be secured by patent is:

1. A method supporting keyword searches of data items in a structured database, the method comprising the computer-implemented steps of:

selecting at least one data item in the structured database, each selected item containing data and each selected item having a corresponding location identifier which identifies the item's location within the structured database;

documenting the selected data items by creating at least one document outside the structured database which contains a textual representation of each selected item's data; and

indexing the documents by creating an index outside the database, the index associating keywords in the textual representation of each selected item's data with that item's location identifier,

wherein the structured database includes data items organized as records in relations according to a data dictionary, the selecting step includes the step of providing a supplemental data dictionary which identifies selected records or tables, and the indexing step only indexes records and tables that are identified by the supplemental data dictionary.

2. The method of claim 1, wherein the indexing step includes providing to a keyword search engine indexing agent both the textual representation of each selected item's data and the selected item's location identifier.

3. The method of claim 2, wherein the indexing agent produces an index that associates keywords with resource locators, and each resource locator includes a textual representation of a data item location identifier.

4. The method of claim 3, wherein the resource locator includes an URL.

5. The method of claim 3, wherein the resource locator includes a file path.

6. The method of claim 3, wherein the textual representations are comprehensive with respect to the data values of selected data items.

7. The method of claim 1, wherein the creating step creates an index containing keywords that are textual representations of data in the selected data items.

8. The method of claim 7, wherein the creating step creates an index containing keywords that are textual representations of non-numeric data in the selected data items.

9. A method supporting keyword searches of data items in a structured database, the method comprising the computer-implemented steps of:

selecting at least one data item in the structured database, each selected item containing data and each selected item having a corresponding location identifier which identifies the item's location within the structured database;

documenting the selected data items by creating at least one document outside the structured database which contains a textual representation of each selected item's data; and

indexing the documents by creating an index outside the database, the index associating keywords in the textual representation of each selected item's data with that item's location identifier,

wherein the indexing step includes providing to a keyword search engine indexing agent both the textual representation of each selected item's data and the selected item's location identifier, the indexing agent produces an index that associates keywords with resource locators, each resource locator includes a textual representation of a data item location identifier, and the resource locator includes a distinguished name.

10. A method supporting keyword searches of data items in a structured database, the method comprising the computer-implemented steps of:

selecting at least one data item in the structured database, each selected item containing data and each selected item having a corresponding location identifier which identifies the item's location within the structured database;

documenting the selected data items by creating at least one document outside the structured database which contains a textual representation of each selected item's data; and

indexing the documents by creating an index outside the database, the index associating keywords in the textual representation of each selected item's data with that item's location identifier,

wherein the creating step creates an index containing keywords that are textual representations of data in the selected data items and also containing every keyword that is a textual representation of data in the selected data items.

11. A method supporting keyword searches of data items in a structured database, the method comprising the computer-implemented steps of:

selecting at least one data item in the structured database, each selected item containing data and each selected item having a corresponding location identifier which identifies the item's location within the structured database;

documenting the selected data items by creating at least one document outside the structured database which contains a textual representation of each selected item's data;

indexing the documents by creating an index outside the database, the index associating keywords in the textual representation of each selected item's data with that item's location identifier; and

logging changes that are made to data items after the creating step and then updating the index to reflect at least some of the changes.

12. A method supporting keyword searches of data items in a structured database, the method comprising the computer-implemented steps of:

selecting at least one data item in the structured database, each selected item containing data and each selected item having a corresponding location identifier which identifies the item's location in the structured database;

allowing a network-roaming indexing agent to create an index which associates keywords with resource locators, each keyword being a textual representation of data from a selected data item and each resource locator containing a textual representation of the corresponding selected item's location identifier;

obtaining a keyword from a search engine interface;

using the index to obtain a resource locator associated with the keyword; and then

using the resource locator to retrieve the item's current data from the structured database.

13. The method of claim 12, wherein the resource locator includes an URL.

14. The method of claim 12, wherein the allowing step reads a data dictionary which identifies only the selected data items.

15. The method of claim 12, wherein the allowing step includes reading data from data items which are records in a relational database.

16. The method of claim 12, wherein the allowing step includes reading data from data items which are nodes in a hierarchical database.

17. The method of claim 12, wherein the allowing step includes reading data from data items which are objects in an object-oriented database.

18. The method of claim 12, wherein the step of using the resource locator comprises extracting a data item's location identifier from the resource locator, and then using the location identifier to retrieve the item's current data.

19. The method of claim 12, wherein the step of using the resource locator includes generating a request to retrieve the item's current data from the database.

20. The method of claim 19, wherein the request includes an SQL query.

21. The method of claim 12, further comprising the computer-implemented step of generating a textual document containing the retrieved data.

22. The method of claim 21, wherein the document is generated in a markup language format.

23. The method of claim 22, wherein the document is generated in HTML format.

24. A computer storage medium having a configuration that represents data and instructions which will cause at least a portion of a computer system to perform method steps for supporting keyword searches of data items in a structured database, the method steps comprising the steps of claim 13.

25. The storage medium of claim 24, wherein the method steps comprise the steps of claim 15.

26. The storage medium of claim 24, wherein the method steps comprise the steps of claim 19.

27. The storage medium of claim 24, wherein the method steps comprise the steps of claim 20.

28. The storage medium of claim 24, wherein the method steps comprise the steps of claim 22.

29. A computer system comprising:

selecting means for selecting data items in a structured database;

retrieving means for retrieving from the database the current data of a selected data item; and

exposing means for exposing to an indexing agent information about a data item's location in the database together with information about the data item's retrieved data,

wherein the structured database includes a relational database, the data items include relational database records or tables, and the selecting means includes a selection data dictionary which specifies only selected relational database records or tables.

30. The system of claim 29, wherein the selecting means includes a schema defining elements of the structured database.

31. The system of claim 29, further comprising an administration tool for modifying the selecting means.

32. The system of claim 31, wherein the selecting means includes a selection data dictionary which specifies only selected relational database records or tables, and the administration tool is capable of creating and modifying the selection data dictionary.

33. The system of claim 29, wherein the retrieving means includes a database reader capable of generating requests to retrieve data from the structured database.

34. The system of claim 33, wherein the database reader is capable of generating SQL queries.

35. The system of claim 29, further comprising the indexing agent.

36. The system of claim 35, wherein the indexing agent includes a web crawler.

37. The system of claim 29, further comprising a search engine interface.

38. The system of claim 37, wherein the search engine interface and the retrieving means reside on different nodes in a network.

39. The system of claim 38, wherein the search engine interface and the retrieving means communicate with one another using a TCP/IP network protocol.

40. The system of claim 38, wherein the search engine interface and the retrieving means communicate with one another using an IPX network protocol.

41. The system of claim 29, further comprising an index produced by the indexing agent.

42. The system of claim 41, wherein the index contains keywords and corresponding resource locators for both the structured database and a textual document information source residing at a different network location than the structured database.

43. The system of claim 41, wherein the index contains keywords and corresponding resource locators for at least two structured databases residing at different network locations.

44. A computer system comprising:

selecting means for selecting data items in a structured database;

retrieving means for retrieving from the database the current data of a selected data item; and

exposing means for exposing to an indexing anent information about a data item's location in the database together with information about the data item's retrieved data, wherein the exposing means includes a page generator capable of generating a textual document containing the retrieved data.

45. The system of claim 44, wherein the page generator is capable of generating an HTML page containing the retrieved data.

* * * * *